ISO 29990:2010

NATIONAL INFOCOMM COMPETENCY FRAMEWORK

NUS National University of Singapore

ISS INSTITUTE OF SYSTEMS SCIENCE

Text Processing using Machine Learning

Deep Learning Foundations

Liling Tan

16 Jan 2019

Overview

Lecture

- **Matrix Calculus for Deep Learning** (120 mins)
  - Scalar derivatives, partial derivatives, vectorized gradients
  - Jacobian Matrix
  - Computation Graph
  - Vector Chain Rule

- **Backpropagation** (45 mins)
  - Jacobian Refresher
  - With Multi-Layered Perceptron

Matrix Calculus for Deep Learning

Prologue

Prologue

- **This lesson is mostly math**
  - Q: "I thought you say we'll see more code than math in your classes"
  - A: "I promise there'll be some code in the hands-on. Also, this will be the one and only lesson with that much math."

- **Why do I have to learn the math? Isn't loss.backward() enough?**
  - "*if you really want to **really understand what's going on under the hood** of these [auto-differentiation] libraries, and **grok academic papers**,… you will need to understand certain bits of the field of matrix calculus*" – *Terence Parr

- **Most of the content in this lesson comes from Terence Parr and Jeremy Howard (2018)** https://explained.ai/matrix-calculus/index.html

*I supposed since there was a reference to Jeremy Howard in 3rd person in sentence before.

Prologue

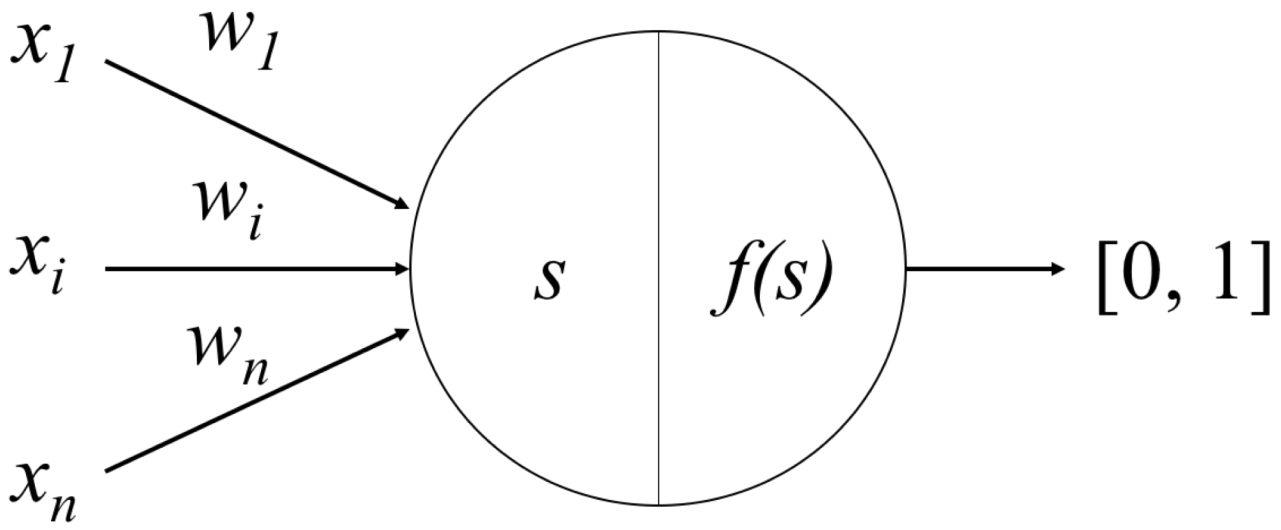
- **This lesson is mostly math, so**

  - Stop me anytime if you have any questions
  - Tell me when I'm going too fast
  - If you don't understand anything on a slide, please ask me to repeat or clarify esp. for this lesson since the formulas will snowball

Matrix Calculus for Deep Learning

Scalar derivatives, partial derivatives, vectorized gradients

Perceptron



Affine function → ***Summation***

$$s = \sum w \cdot x$$

Transformation

$$f(s) = \frac{1}{1+e^{-s}}$$

← **Activation function**

Manual Differentiation

```
import math
import numpy as np
np.random.seed(0)

def sigmoid(x): # Returns values that sums to one.
    return 1 / (1 + np.exp(-x))

def sigmoid_derivative(sx):
    # See https://math.stackexchange.com/a/1225116
    return sx * (1 - sx)
```

Manual Differentiation

Let's denote the sigmoid function as $\sigma(x) = \frac{1}{1+e^{-x}}$.

$$
\begin{aligned}
\frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left[\frac{1}{1+e^{-x}}\right] \\
&= \frac{d}{dx}\left(1+\mathrm{e}^{-x}\right)^{-1} \\
&= -(1+e^{-x})^{-2}(-e^{-x}) \\
&= \frac{e^{-x}}{(1+e^{-x})^2} \\
&= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \cdot \frac{(1+e^{-x})-1}{1+e^{-x}} \\
&= \frac{1}{1+e^{-x}} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \\
&= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) \\
&= \sigma(x) \cdot (1-\sigma(x))
\end{aligned}
$$

Refresher:

- Calculus: https://www.khanacademy.org/math/differential-calculus
- ML Cheatsheet: https://ml-cheatsheet.readthedocs.io/en/latest/calculus.html
- Math Cheatsheet: http://tutorial.math.lamar.edu/pdf/Calculus_Cheat_Sheet_Derivatives.pdf

Manual Differentiation

Rule	$f(x)$	$\frac{d}{dx}f(x)$	Example
Constant	c	0	$\frac{d}{dx}99 = 0$
Multiply by Constant	cf	$c\frac{df}{dx}$	$\frac{d}{dx}3x = 3$
Power Rule	x^n	nx^{n-1}	$\frac{d}{dx}x^3 = 3x^2$
Sum Rule	$f + g$	$\frac{df}{dx} + \frac{dg}{dx}$	$\frac{d}{dx}(x^4 + 3x) = 4x^3 + 3$
Difference Rule	$f - g$	$\frac{df}{dx} - \frac{dg}{dx}$	$\frac{d}{dx}(x^4 - 3x) = 4x^3 - 3$
Product Rule	fg	$f\frac{dg}{dx} + g\frac{df}{dx}$	$\frac{d}{dx}(x^3x + 3x)(2x^2 - 1)$ $= (9x^2 + 3) + (6x)$
Chain Rule	$f(g(x))$	$\frac{df(u)}{du}\frac{du}{dx}$, where $u = g(x)$	$\frac{d}{dx}\ln(x^2) = \frac{1}{x^2}2x = \frac{1}{x}$

Adapted from Parr and Howard (2018)

Scalar Derivatives

$$f(u) = \ln(u)$$
$$g(x) = x^2$$
$$f(g(x)) = \ln(x^2)$$

$$y = \ln(x)$$
$$e^y = x$$
$$e^y \frac{dy}{dx} = 1$$
$$x \frac{dy}{dx} = 1$$
$$\frac{dy}{dx} = \frac{1}{x}$$

Definitions

- Think of ***scalar derivatives*** $\frac{d}{dx}$ as an ***operator that maps a single variable/parameter to a function*** $f(x)$ with respect to (w.r.t.) x

Scalar Derivatives

$$f(u) = \ln(u)$$
$$g(x) = x^2$$
$$f(g(x)) = \ln(x^2)$$

$$y = \ln(x^2)$$
$$e^y = x^2$$
$$e^y \frac{dy}{dx} = 2x$$
$$x^2 \frac{dy}{dx} = 2x$$
$$\frac{dy}{dx} = \frac{1}{x^2} * 2x$$
$$= \frac{2}{x}$$

Chain Rule

$$f(u) = \ln(u)$$
$$g(x) = x^2$$
$$f(g(x)) = \ln(x^2)$$
$$\frac{d}{du}\ln(u) = \frac{1}{u}$$

$$\frac{d}{dx}f(g(x)) = \frac{df(u)}{du}\frac{du}{dx}$$
$$= \frac{1}{x^2}\frac{d}{dx}g(x)$$
$$= \frac{1}{x^2}\,2x$$
$$= \frac{2}{x}$$

Chain Rule

$$f(u) = 2(u)^3$$

$$\mathrm{u} = g(x) = (x^2 + 9)$$

$$f(g(x)) = 2(x^2 + 9)^3$$

$$\frac{df(u)}{du} = 6u^2 = 6(x^2 + 9)^2$$

$$\frac{du}{dx} = 2x$$

$$\frac{d}{dx} f(g(x))$$

$$= \frac{df(u)}{du} \frac{du}{dx}$$

$$= 6(x^2 + 9)^2 . 2x$$

Definitions

- Think of ***scalar derivatives*** $\frac{d}{dx}$ as an ***operator that maps a single variable/parameter to a function*** $f(x)$ with respect to (w.r.t.) x

- The ***partial derivatives*** e.g. $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ ***"individualizes" the operations to map multiple variable/parameter to the multi-variable function*** , e.g. $f(x, y)$

Partial Derivatives

$$f(x, y) = 3x^2(2y + 7)$$

$$\frac{\partial}{\partial x} 3x^2(2y + 7) = (2y + 7)\frac{\partial}{\partial x} 3x^2 = 6x(2y + 7)$$

$$\frac{\partial}{\partial y} 3x^2(2y + 7) = 3x^2 \frac{\partial}{\partial y}(2y + 7) = 6x^2$$

Definitions

- Think of ***scalar derivatives*** $\frac{d}{dx}$ as an ***operator that maps a single variable/parameter to a function*** $f(x)$ with respect to (w.r.t.) x

- The ***partial derivatives*** e.g. $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ ***"individualizes" the operations to map multiple variable/parameter to the multi-variable function*** , e.g. $f(x, y)$

- In vector calculus, we pack ***partial derivatives as a vector*** and ***represent the gradient of a function*** $f(x, y)$ as $\nabla f(x, y) = \left[\frac{\partial}{\partial x} f(x, y), \frac{\partial}{\partial y} f(x, y)\right]$

Vectorized Gradient

$$f(x,y) = 3x^2(2y+7)$$

$$\frac{\partial}{\partial x} 3x^2(2y+7) = (2y+7)\frac{\partial}{\partial x} 3x^2 = 6x(2y+7)$$

$$\frac{\partial}{\partial y} 3x^2(2y+7) = 3x^2 \frac{\partial}{\partial y}(2y+7) = 6x^2$$

$$\nabla f(x,y) = \left[\frac{\partial}{\partial x} f(x,y), \frac{\partial}{\partial y} f(x,y)\right]$$

$$= [6x(2y+7), 6x^2]$$

Notations

Math	*aka	Math-sy English	Human-ish English
$\frac{dy}{dx}$	$f'(x)$	Scalar derivatives	Calculates changes for one variable
	$d_x f$		
	$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$		
$\frac{\partial y}{\partial x}$	$f'(x)$	Partial derivatives	Calculates changes for multiple variable in parts
	$\partial_x f$		
	$\lim_{h \to 0} \frac{f(x_i, \ldots, x_i + h, \ldots, x_n) - f(x_i, \ldots, x_n)}{h}$		
$\nabla f(x, y)$		Vectorized gradients	Calculates changes for multiple variables in parts in a vectorized manner

Matrix Calculus for Deep Learning

Jacobian matrix

Jacobian Matrix

- When computing ***partial derivatives for multiple functions***, we organize them in a ***Jacobian matrix*** where each row contain the gradients of each function, e.g.

Jacobian Matrix

$$f(x,y) = 3x^2(2y+7) \qquad \nabla f(x,y) = [6x(2y+7) \quad 6x]$$

$$g(x,y) = 4x + y^2 \qquad \nabla g(x,y) = [4 \quad 2y]$$

$$J = \begin{bmatrix} \nabla f(x,y) \\ \nabla g(x,y) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x,y) & \frac{\partial}{\partial y} f(x,y) \\ \frac{\partial}{\partial x} g(x,y) & \frac{\partial}{\partial y} g(x,y) \end{bmatrix} = \begin{bmatrix} 6x(2y+7) & 6x \\ 4 & 2y \end{bmatrix}$$

Generalization of Jacobian

Consider $\boldsymbol{n}$ no. of parameters as a single $\boldsymbol{X}$ vector:

$$X = [x_1 \quad x_2 \quad \cdots \quad x_n]$$

Let $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{X})$ be the single function that takes in an $\boldsymbol{X}$ vector and assume $\boldsymbol{m}$ no. of functions

$$Y = \begin{array}{l} y_1 = f_1(X) \\ y_2 = f_2(X) \\ \vdots \\ y_m = f_m(X) \end{array}$$

Generalization of Jacobian

$\boldsymbol{n}$ parameters as $\boldsymbol{X}$ vector: $X = [x_1 \quad x_i \quad \cdots \quad x_n]$

$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{X})$ takes $\boldsymbol{X}$ vector with $\mathbf{m}$ functions:

$$Y = \begin{matrix} y_1 = f_1(X) \\ y_j = f_j(X) \\ \vdots \\ y_m = f_m(X) \end{matrix}$$

Consider an identify function where $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{X}) = \boldsymbol{X}$ with $\mathbf{n}$ functions where each function maps to a respective $\boldsymbol{x_i}$:

$$Y = \begin{matrix} y_1 = f_1(X) = x_1 \\ y_j = f_j(X) = x_i \\ \vdots \\ y_n = f_n(X) = x_n \end{matrix}$$

Generalization of Jacobian

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \nabla f_1(\boldsymbol{X}) \\ \nabla f_2(\boldsymbol{X}) \\ \vdots \\ \nabla f_m(\boldsymbol{X}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f_1(\boldsymbol{X}) \\ \frac{\partial}{\partial x} f_2(\boldsymbol{X}) \\ \vdots \\ \frac{\partial}{\partial x} f_m(\boldsymbol{X}) \end{bmatrix}$$

Generalization of Jacobian

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x} f_1(\boldsymbol{X}) \\ \frac{\partial}{\partial x} f_2(\boldsymbol{X}) \\ \vdots \\ \frac{\partial}{\partial x} f_m(\boldsymbol{X}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(\boldsymbol{X}) & \frac{\partial}{\partial x_2} f_1(\boldsymbol{X}) & \cdots & \frac{\partial}{\partial x_n} f_1(\boldsymbol{X}) \\ \frac{\partial}{\partial x_1} f_2(\boldsymbol{X}) & \frac{\partial}{\partial x_2} f_2(\boldsymbol{X}) & \cdots & \frac{\partial}{\partial x_n} f_2(\boldsymbol{X}) \\ & & \vdots & \\ \frac{\partial}{\partial x_1} f_m(\boldsymbol{X}) & \frac{\partial}{\partial x_2} f_m(\boldsymbol{X}) & \cdots & \frac{\partial}{\partial x_n} f_m(\boldsymbol{X}) \end{bmatrix}$$

Jacobian of Identity Function

Identity Function: $f_m(\boldsymbol{X}) = \boldsymbol{X}$, where $m = n$

$$\boldsymbol{f}(\boldsymbol{X}) = \begin{bmatrix} f_1(\boldsymbol{X}) = x_1 \\ f_2(\boldsymbol{X}) = x_2 \\ \vdots \\ f_m(\boldsymbol{X}) = x_n \end{bmatrix}$$

Jacobian of Identity Function

Identity Function: $f_m(\boldsymbol{X}) = \boldsymbol{X}$, where $m = n$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x} f_1(\boldsymbol{X}) \\ \frac{\partial}{\partial x} f_2(\boldsymbol{X}) \\ \vdots \\ \frac{\partial}{\partial x} f_m(\boldsymbol{X}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} x_1 & \frac{\partial}{\partial x_2} x_1 & \cdots & \frac{\partial}{\partial x_n} x_1 \\ \frac{\partial}{\partial x_1} x_2 & \frac{\partial}{\partial x_2} x_2 & \cdots & \frac{\partial}{\partial x_n} x_2 \\ & & \vdots & \\ \frac{\partial}{\partial x_1} x_n & \frac{\partial}{\partial x_2} x_n & \cdots & \frac{\partial}{\partial x_n} x_n \end{bmatrix}$$

Jacobian of Identity Function

Identity Function: $f_m(\boldsymbol{X}) = \boldsymbol{X}$, where $m = n$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x} f_1(\boldsymbol{X}) \\ \frac{\partial}{\partial x} f_2(\boldsymbol{X}) \\ \vdots \\ \frac{\partial}{\partial x} f_m(\boldsymbol{X}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} x_1 & 0 & \dots & 0 \\ 0 & \frac{\partial}{\partial x_2} x_2 & \dots & 0 \\ 0 & 0 & \dots & \frac{\partial}{\partial x_n} x_n \end{bmatrix}$$

Jacobian of Identity Function

Identity Function: $f_m(\boldsymbol{X}) = \boldsymbol{X}$, where $m = n$

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x} f_1(\boldsymbol{X}) \\ \frac{\partial}{\partial x} f_2(\boldsymbol{X}) \\ \vdots \\ \frac{\partial}{\partial x} f_m(\boldsymbol{X}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ 0 & 0 & \dots & 1 \end{bmatrix}$$

Derivatives of Vectors with Element-wise Binary Operators

Let $\bigcirc$ represent any "element-wise scalar operations",

Consider the **functions f and g** that maps the **inputs w and x** through an element-wise scalar operations to y

$$y = f_m(w_{1\ldots n}) \bigcirc g_m(x_{1\ldots n})$$

Derivatives of Vectors with Element-wise Binary Operators

For example the ***affine function:***

$$s = \sum_{i}^{n} w \cdot x$$

We iterate through $i \ldots n$ element-wise and multiply $w_i x_i$
we can represent it with the notation from the previous slide

$$s = f(w) \otimes \ g(x)$$

In the affine function, there isn't any additional function on the w and x vectors , so it can be simplified to $s = w \otimes \ x$

Derivatives of Vectors with Element-wise Binary Operators

Consider the **functions f and g** that maps the **inputs w and x** through an element-wise scalar operations to y

$$y = f_m(w_{1\ldots n}) \bigcirc g_m(x_{1\ldots n}) \quad \text{, where } m = n = |w| = |x|$$

Going back a few slides, and apply partial derivatives to the y equation above:

Derivatives of Vectors with Element-wise Binary Operators

$$y = f_m(w_{1\ldots n}) \bigcirc g_m(x_{1\ldots n}) \quad \text{, where } m = n = |w| = |x|$$

Apply ***partial derivatives*** to the y equation **w.r.t.** $\boldsymbol{x}$:

$$\frac{\partial y}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial}{\partial x} f_1(W) \bigcirc g_1(X) \\ \frac{\partial}{\partial x} f_2(W) \bigcirc g_2(X) \\ \vdots \\ \frac{\partial}{\partial x} f_n(W) \bigcirc g_n(X) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(W) \bigcirc g_1(X) & \ldots & \frac{\partial}{\partial x_n} f_1(W) \bigcirc g_1(X) \\ \frac{\partial}{\partial x_1} f_2(W) \bigcirc g_2(X) & \ldots & \frac{\partial}{\partial x_n} f_2(W) \bigcirc g_2(X) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_1} f_n(W) \bigcirc g_n(X) & \ldots & \frac{\partial}{\partial x_n} f_n(W) \bigcirc g_n(X) \end{bmatrix}$$

Derivatives of Vectors with Element-wise Binary Operators

$$y = f_m(w_{1\ldots n}) \bigcirc g_m(x_{1\ldots n}) \quad \text{, where } m = n = |w| = |x|$$

Apply ***partial derivatives*** to the y equation **w.r.t. w** :

$$\frac{\partial y}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{\partial}{\partial w} f_1(W) \bigcirc g_1(X) \\ \frac{\partial}{\partial w} f_2(W) \bigcirc g_2(X) \\ \vdots \\ \frac{\partial}{\partial w} f_n(W) \bigcirc g_n(X) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial w_1} f_1(W) \bigcirc g_1(X) & \ldots & \frac{\partial}{\partial w_n} f_1(W) \bigcirc g_1(X) \\ \frac{\partial}{\partial w_1} f_2(W) \bigcirc g_2(X) & \ldots & \frac{\partial}{\partial w_n} f_2(W) \bigcirc g_2(X) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial w_1} f_n(W) \bigcirc g_n(X) & \ldots & \frac{\partial}{\partial w_n} f_n(W) \bigcirc g_n(X) \end{bmatrix}$$

Derivatives of Vectors with Element-wise Binary Operators

$$y = f_m(w_{1\ldots n}) \bigcirc g_m(x_{1\ldots n}) \quad \text{, where } m = n = |w| = |x|$$

Very often the Jacobian is zero everywhere except the diagonal, like what we see in the identity matrix before.

$$\frac{\partial y}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{\partial}{\partial w_1} f_1(W) \bigcirc g_1(X) & 0 & 0 & 0 \\ 0 & \frac{\partial}{\partial w_2} f_2(W) \bigcirc g_2(X) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial w_n} f_n(W) \bigcirc g_n(X) \end{bmatrix}$$

Derivatives of Vectors with Element-wise Binary Operators

To convince ourselves, lets take a look at the + operator,

$$\mathrm{y} = f(w) + g(x)$$

as a scalar equation:

$$y_i = f_i(w) + g_i(x)$$

gets reduce to:

$$y_i = f_i(w_i) + g_i(x_i) = w_i + x_i$$

$$\frac{\partial}{\partial w_i}(w_i + x_i) = 1 + 0 = 1$$

$$\frac{\partial}{\partial w_i}(w_j + x_k) = 0 + 0 = 0$$

$$\frac{\partial}{\partial x_i}(w_i + x_i) = 0 + 1 = 1$$

$$\frac{\partial}{\partial x_i}(w_j + x_k) = 0 + 0 = 0$$

Derivatives of Vectors with Element-wise Binary Operators

Lets take a look at ***affine function w/o the sum*** with the $\otimes$ operator,

$$\mathrm{y} = f(w) \otimes g(x)$$

as a scalar equation:

$$y_i = f_i(w) \otimes g_i(x)$$

gets reduce to:

$$y_i = f_i(w_i) \otimes g_i(x_i)$$

$$\frac{\partial}{\partial w_i}(w_i \otimes x_i) = \mathbb{R} + 0 = \mathbb{R}$$

$$\frac{\partial}{\partial w_i}(w_j \otimes x_k) = 0 + 0 = 0$$

$$\frac{\partial}{\partial x_i}(w_i \otimes x_i) = 0 + \mathbb{R} = \mathbb{R}$$

$$\frac{\partial}{\partial x_i}(w_j \otimes x_k) = 0 + 0 = 0$$

Derivatives of Vectors with Element-wise Binary Operators

To simplify this, **w.r.t. w**:

$$\frac{\partial y}{\partial w} = \begin{bmatrix} \frac{\partial}{\partial w_1} f_1(W) \bigcirc g_1(X) & 0 & 0 & 0 \\ 0 & \frac{\partial}{\partial w_2} f_2(W) \bigcirc g_2(X) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial w_n} f_n(W) \bigcirc g_n(X) \end{bmatrix}$$

We could rewrite it as such:

$$\frac{\partial y}{\partial w} = diag(\frac{\partial}{\partial w_1} f_1(W) \bigcirc g_1(X), \frac{\partial}{\partial w_2} f_2(W) \bigcirc g_2(X), \ldots, \frac{\partial}{\partial w_n} f_n(W) \bigcirc g_n(X))$$

Derivatives of Vectors with Element-wise Binary Operators

Similarly **w.r.t. x** :

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(W) \bigcirc g_1(X) & 0 & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} f_2(W) \bigcirc g_2(X) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial x_n} f_n(W) \bigcirc g_n(X) \end{bmatrix}$$

We could rewrite it as such:

$$\frac{\partial y}{\partial x} = diag(\frac{\partial}{\partial x_1} f_1(W) \bigcirc g_1(X), \frac{\partial}{\partial x_2} f_2(W) \bigcirc g_2(X), \ldots, \frac{\partial}{\partial x_n} f_n(W) \bigcirc g_n(X))$$

Special Cases for Jacobian Partial Derivatives

Op	Partial with respect to $\mathbf{w}$
$+$	$\frac{\partial(\mathbf{w}+\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i+x_i)}{\partial w_i} \ldots) = diag(\vec{1}) = I$
$-$	$\frac{\partial(\mathbf{w}-\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i-x_i)}{\partial w_i} \ldots) = diag(\vec{1}) = I$
$\otimes$	$\frac{\partial(\mathbf{w}\otimes\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i\times x_i)}{\partial w_i} \ldots) = diag(\mathbf{x})$
$\oslash$	$\frac{\partial(\mathbf{w}\oslash\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i/x_i)}{\partial w_i} \ldots) = diag(\ldots \frac{1}{x_i} \ldots)$

Op	Partial with respect to $\mathbf{x}$
$+$	$\frac{\partial(\mathbf{w}+\mathbf{x})}{\partial \mathbf{x}} = I$
$-$	$\frac{\partial(\mathbf{w}-\mathbf{x})}{\partial \mathbf{x}} = diag(\ldots \frac{\partial(w_i-x_i)}{\partial x_i} \ldots) = diag(-\vec{1}) = -I$
$\otimes$	$\frac{\partial(\mathbf{w}\otimes\mathbf{x})}{\partial \mathbf{x}} = diag(\mathbf{w})$
$\oslash$	$\frac{\partial(\mathbf{w}\oslash\mathbf{x})}{\partial \mathbf{x}} = diag(\ldots \frac{-w_i}{x_i^2} \ldots)$

(Parr and Howard, 2018)

Jacobian Matrix: Generalization of the Gradient

Given a function ***f(x)*** with ***m*** **outputs** and ***n*** **inputs**:

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(x_1, x_2, \ldots, x_n), \ldots, f_m(x_1, x_2, \ldots, x_n)]$$

Jacobian:

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Inside the Jacobian:

$$\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)_{ij} = \frac{\partial f_i}{\partial x_j}$$

Matrix Calculus for Deep Learning

Computation Graph

Chain Rule again…

To solve derivatives of y w.r.t. x :

$$y = f(x) = \ln(\sin(x^3)^2)$$

1. Introduce intermediate variables

$$\begin{aligned} u_1 &= f_1(x) &&= x^3 \\ u_2 &= f_2(u_1) &&= \sin(u_1) \\ u_3 &= f_3(u_2) &&= {u_2}^2 \\ y = u_4 &= f_4(u_3) &&= \ln(u_3) \end{aligned}$$

Chain Rule again…

1. Introduce intermediate variables

2. Compute derivatives of intermediate variables

$$\frac{d}{du_1}u_1 = \frac{d}{dx}x^3 = 3x^2$$

$$\frac{d}{du_2}u_2 = \frac{d}{dx}\sin(u_1) = \cos(u_1)$$

$$\frac{d}{du_3}u_3 = \frac{d}{dx}{u_2}^2 = 2u_2$$

$$\frac{d}{du_4}u_4 = \frac{d}{dx}\ln(u_3) = 1/u_3$$

Chain Rule again…

1. Introduce intermediate variables

2. Compute derivatives of intermediate variables

3. Combine intermediate derivatives

$$\frac{dy}{dx} = \frac{du_4}{dx} = \frac{du_4}{du_3}\frac{du_3}{du_2}\frac{du_2}{du_1}\frac{du_1}{dx} = 3x^2 \cos(u_1)\, 2u_2\, {u_3}^{-1}$$

Chain Rule again…

1. **Introduce intermediate variables**
2. **Compute derivatives of intermediate variables**
3. **Combine intermediate derivatives**
4. **Substitute back the u and simplify**

$$\frac{dy}{dx} = 3x^2 \cos(u_1)\, 2u_2\, {u_3}^{-1}$$

$$= \frac{3x^2 \cos(x^3)\, 2sin(x^3)}{sin(x^3)^2}$$

$$= \frac{6x^2 \cos(x^3)}{sin(x^3)}$$

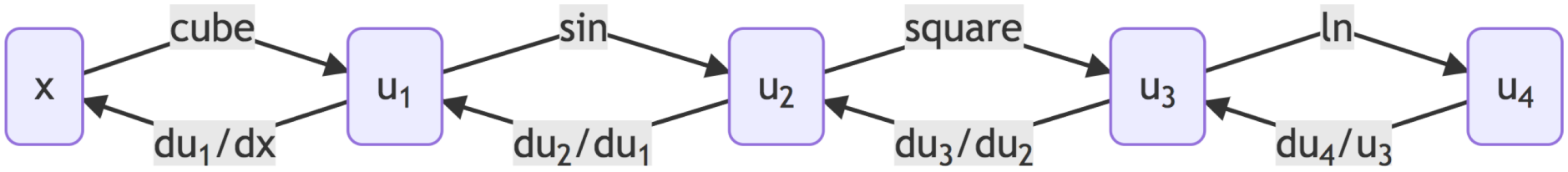
Derivatives and Computation Graph

Lets go back to the intermediate functions

$$
\begin{array}{rcccc}
 & u_1 & = & cube(x) & = & x^3 \\
 & u_2 & = & sin(x) & = & \sin(u_1) \\
 & u_3 & = & square(x) & = & {u_2}^2 \\
y & = u_4 & = & ln(x) & = & \ln(u_3)
\end{array}
$$

$$
\frac{dy}{dx} = \frac{du_4}{dx} = \frac{du_4}{du_3}\frac{du_3}{du_2}\frac{du_2}{du_1}\frac{du_1}{dx}
$$

When the input ***x flows in a single direction*** to the end to produce $\boldsymbol{y}$, we can simply ***"flow" backwards in derivatives of the chain of operations*** to differential.



Partial Derivatives thru Computation Graph

To solve derivatives of y w.r.t. x :

$$y = f(x) = xe^{x^2}$$

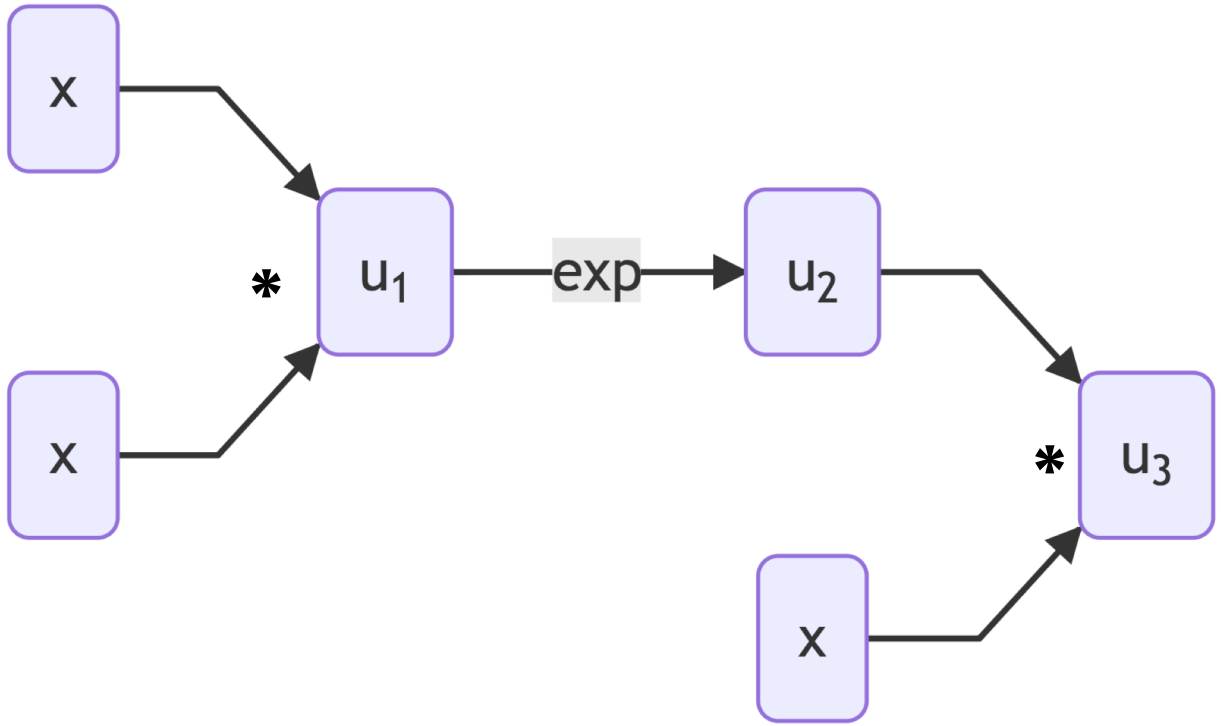
1. Introduce intermediate variables

$$\begin{aligned} u_1 &= f_1(x) &&= x^2 \\ u_2 &= f_2(x) &&= \exp(u_1) \\ y = u_3 &= f_3(x) &&= xu_2 \end{aligned}$$

Partial Derivatives thru Computation Graph

$$
\begin{aligned}
 & u_1 = f_1(x) = x^2 \\
 & u_2 = f_2(x) = \exp(u_1) \\
y = & u_3 = f_3(x) = xu_2
\end{aligned}
$$

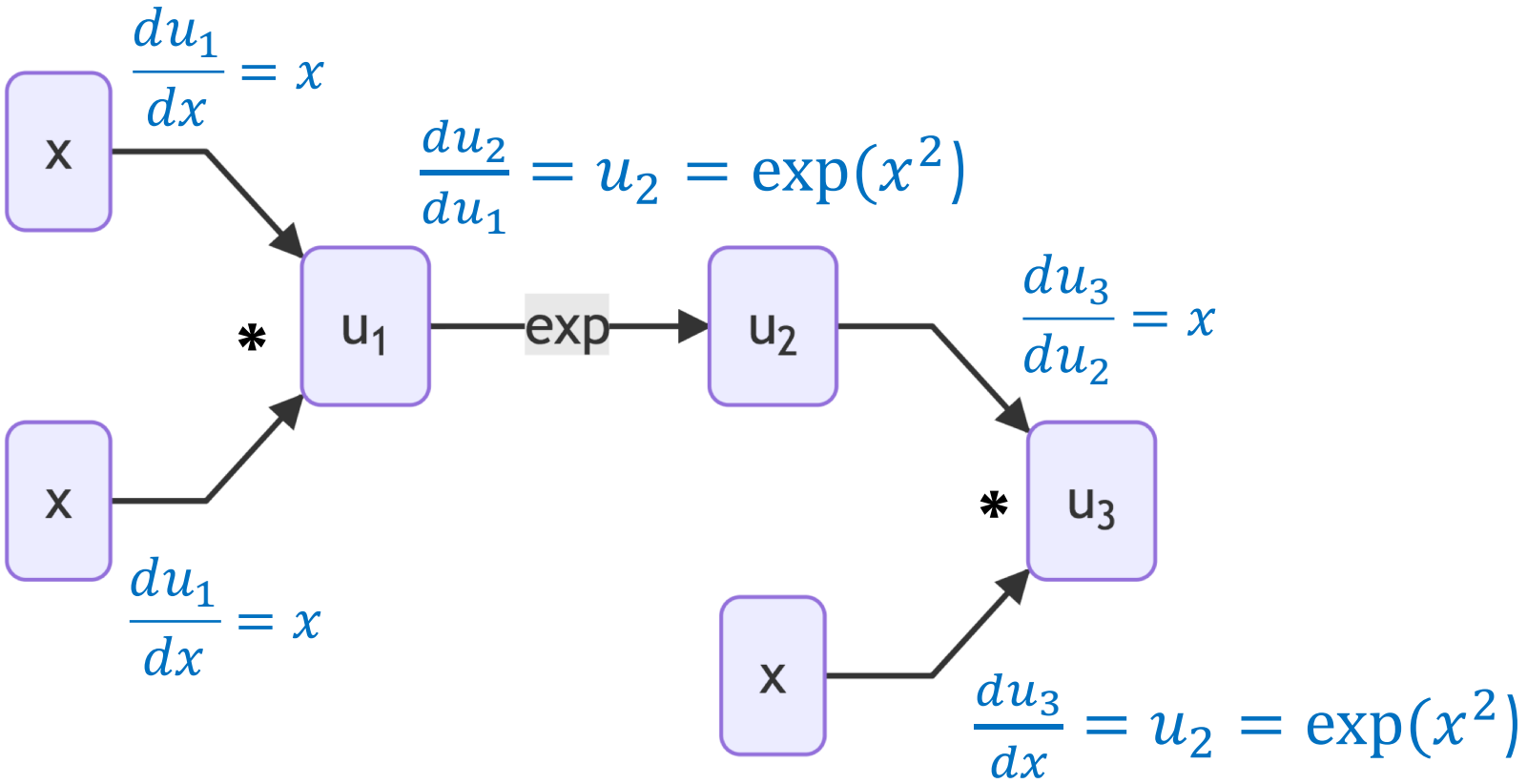
First, draw the computation graph, layout the data flow and ***separate every instance of x as individual node***.



Partial Derivatives thru Computation Graph

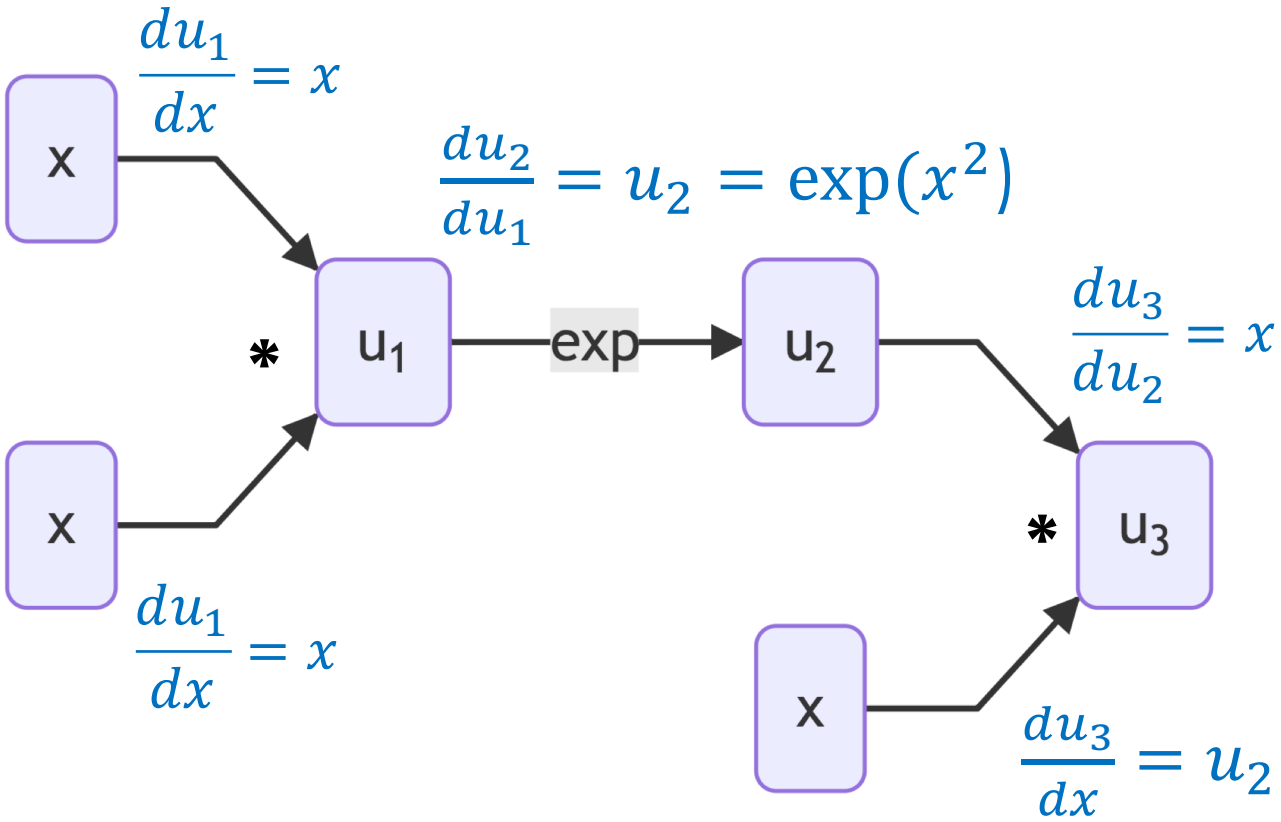
$$
\begin{array}{rcccc}
 & u_1 & = & f_1(x) & = & x^2 \\
 & u_2 & = & f_2(x) & = & \exp(u_1) \\
y = & u_3 & = & f_3(x) & = & xu_2
\end{array}
$$

Compute all the ***derivatives all edges***



Partial Derivatives thru Computation Graph

$$
\begin{aligned}
 & u_1 = f_1(x) = x^2 \\
 & u_2 = f_2(x) = \exp(u_1) \\
y = & u_3 = f_3(x) = xu_2
\end{aligned}
$$

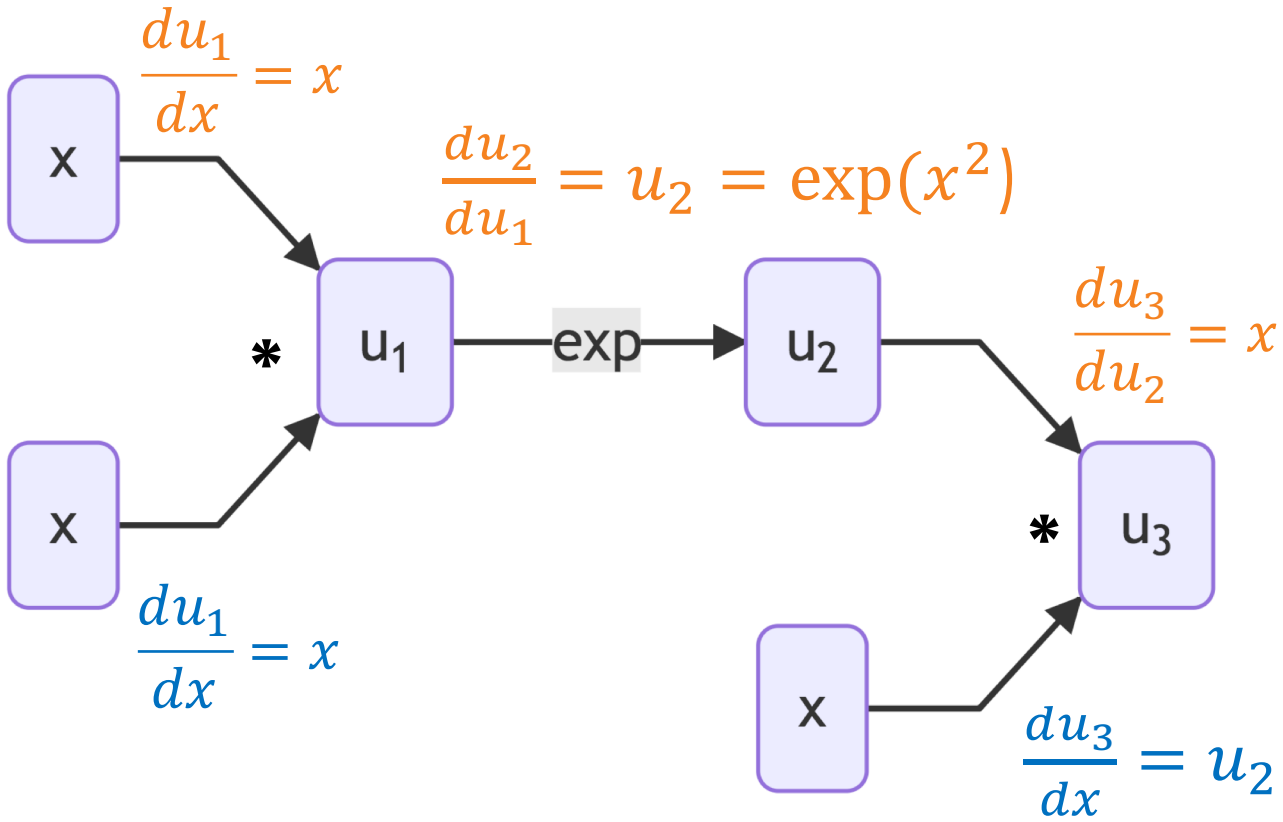


To compute the partial derivative of $\frac{\partial y}{\partial x}$

go backwards from the last node to every x node, multiply all intermediate the derivatives on the edges

Partial Derivatives thru Computation Graph

$$
\begin{array}{ccccccc}
 & & u_1 & = & f_1(x) & = & x^2 \\
 & & u_2 & = & f_2(x) & = & \exp(u_1) \\
y & = & u_3 & = & f_3(x) & = & xu_2
\end{array}
$$



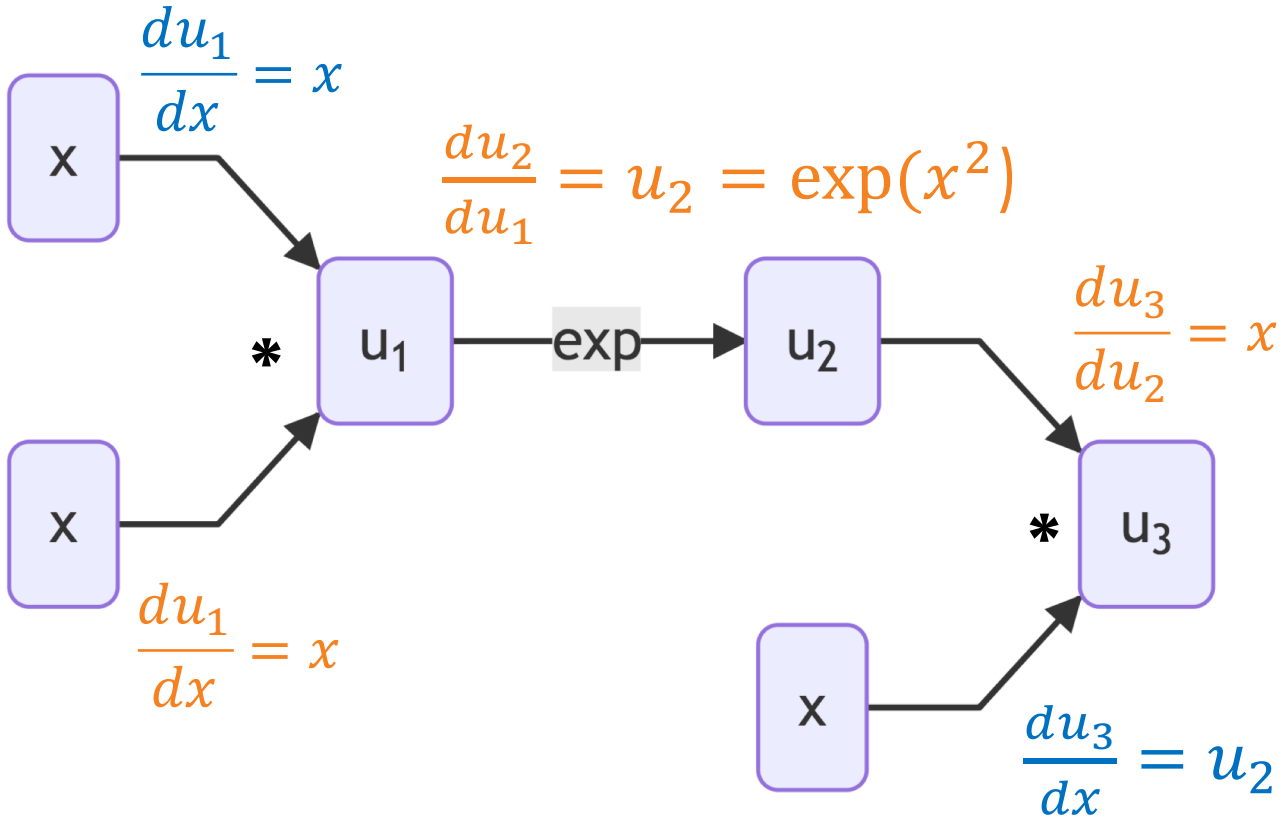
To compute the partial derivative of $\frac{\partial y}{\partial x}$

go backwards from the last node to every x node, multiply all intermediate the derivatives on the edges

$$\frac{\partial y}{\partial x} = x.\exp(x^2).x$$

Partial Derivatives thru Computation Graph

$$
\begin{array}{rcccc}
 & u_1 & = & f_1(x) & = & x^2 \\
 & u_2 & = & f_2(x) & = & \exp(u_1) \\
y & = u_3 & = & f_3(x) & = & xu_2
\end{array}
$$



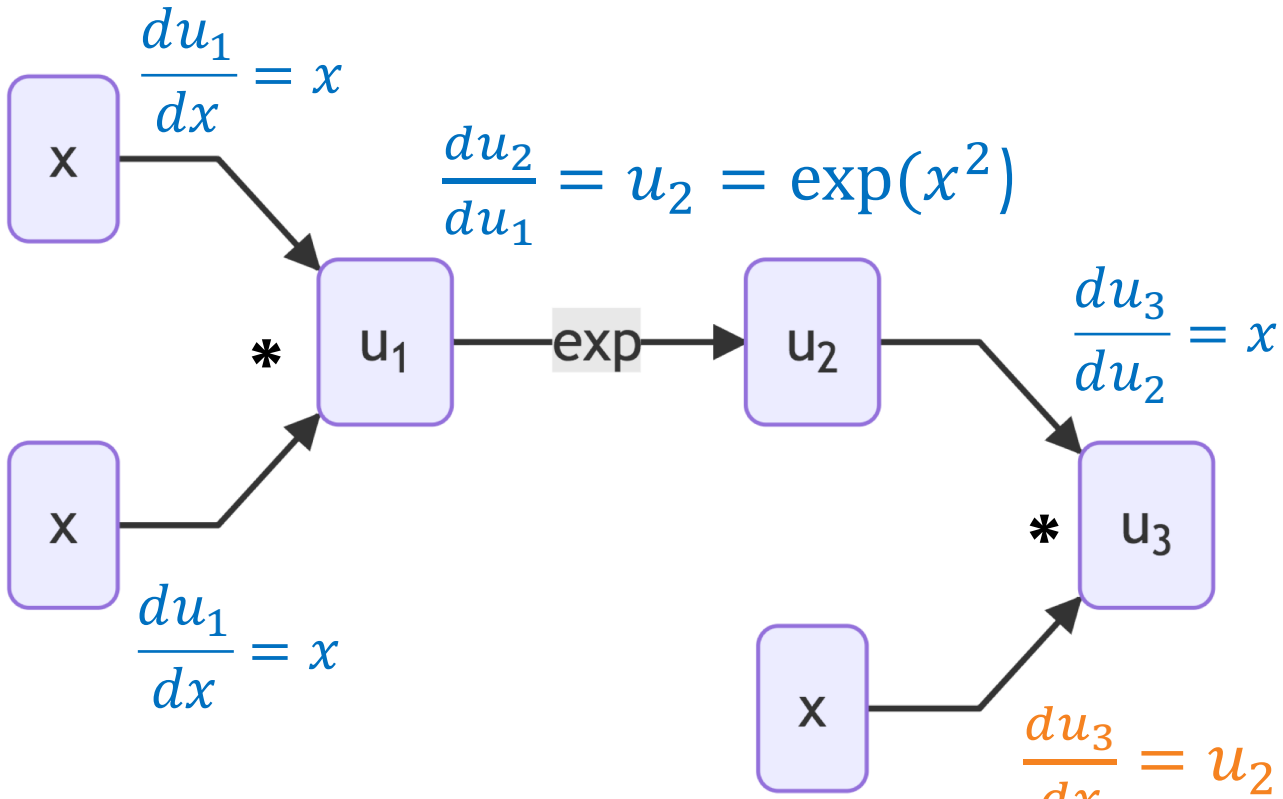
To compute the partial derivative of $\frac{\partial y}{\partial x}$

go backwards from the last node to every x node, multiply all intermediate the derivatives on the edges

$$\frac{\partial y}{\partial x} = x.\exp(x^2).x + x.\exp(x^2).x$$

Partial Derivatives thru Computation Graph

$$
\begin{aligned}
 & u_1 = f_1(x) = x^2 \\
 & u_2 = f_2(x) = \exp(u_1) \\
y = & u_3 = f_3(x) = xu_2
\end{aligned}
$$



To compute the partial derivative of $\frac{\partial y}{\partial x}$

go backwards from the last node to every x node, multiply all intermediate the derivatives on the edges and sum them

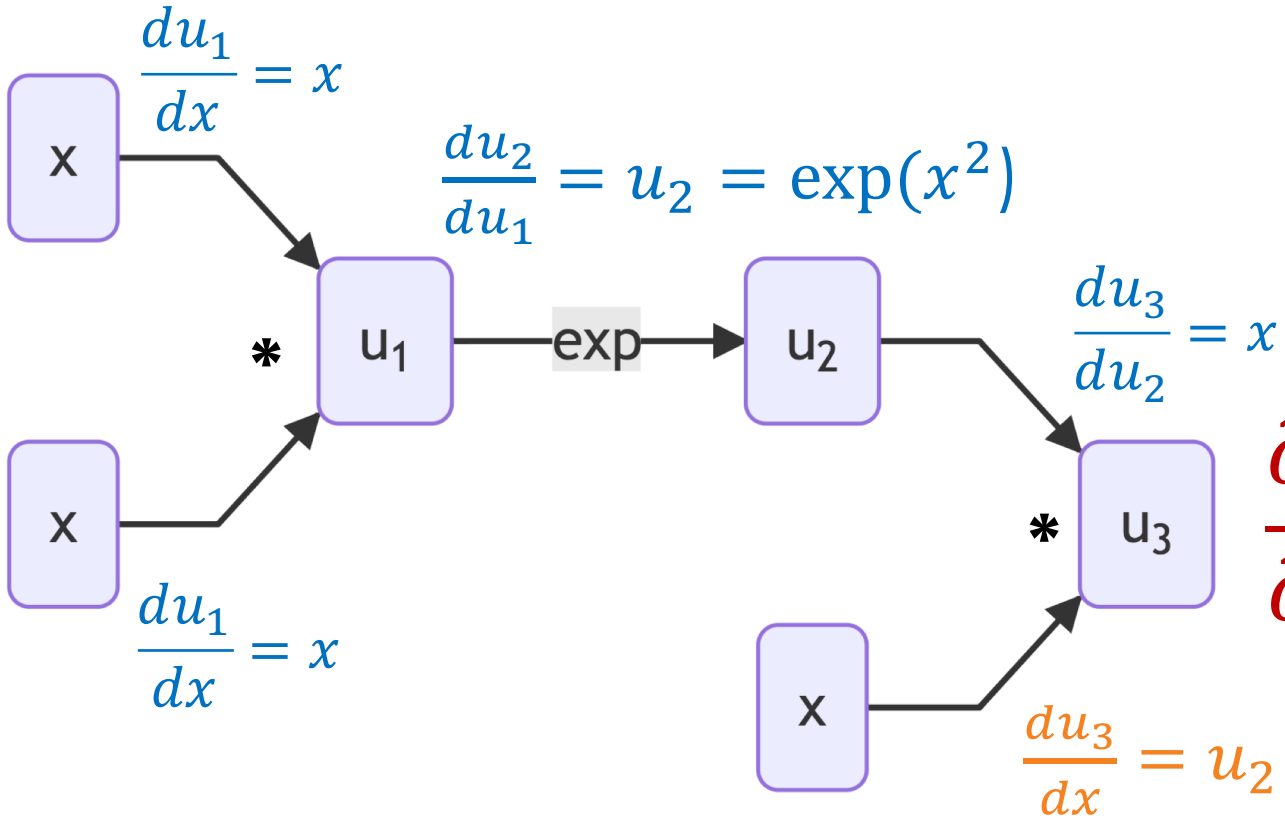
$$
\begin{aligned}
\frac{\partial y}{\partial x} = & x.\exp(x^2).x \\
& + x.\exp(x^2).x \\
& + \exp(x^2)
\end{aligned}
$$

Partial Derivatives thru Computation Graph

$$
\begin{aligned}
 & u_1 = f_1(x) = x^2 \\
 & u_2 = f_2(x) = \exp(u_1) \\
y = & u_3 = f_3(x) = xu_2
\end{aligned}
$$

To compute the partial derivative of $\frac{\partial y}{\partial x}$

go backwards from the last node to every x node, multiply all intermediate the derivatives on the edges and sum them



$$\frac{\partial y}{\partial x} = \exp(x^2).(2x^2 + 1)$$

Sanity Check

To solve derivatives of y w.r.t. x :

$$y = f(x) = xe^{x^2}$$

Apply product rule:

$$\frac{\partial y}{\partial x} = x\frac{\partial e^{x^2}}{\partial x} + e^{x^2}\frac{\partial x}{\partial x}$$

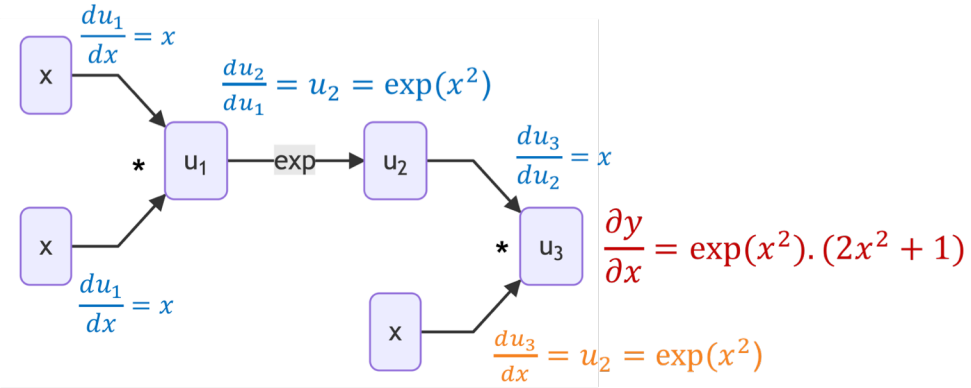
$$= x.2x.e^{x^2} + e^{x^2}.1$$

$$= e^{x^2}(2x^2 + 1)$$

Total Derivative Chain Rule

To simplify the formulation of partial derivatives $\frac{\partial y}{\partial x}$ *we can say that the "total" partial derivatives is*

$$\frac{\partial y}{\partial x} = \frac{\partial f(x, u_1, \ldots, u_n)}{\partial x} = \frac{\partial u_3(u_2, x)}{\partial x}$$



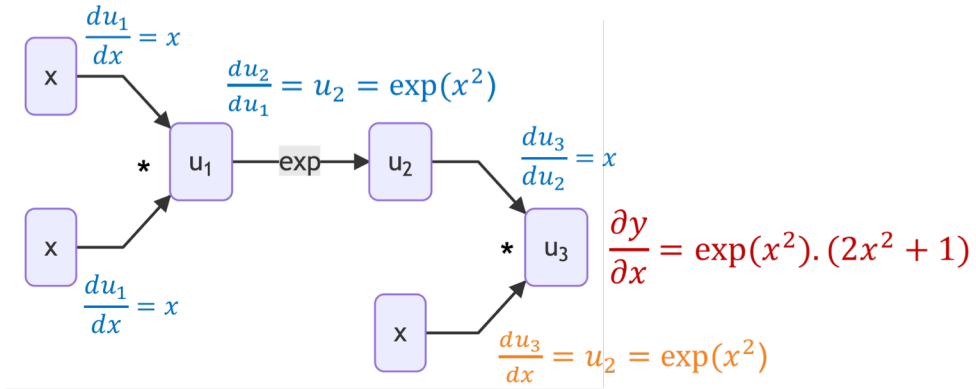
$$= \frac{\partial u_3}{\partial x} + \frac{\partial u_3}{\partial u_2}\frac{\partial u_2}{\partial x} + \frac{\partial u_3}{\partial u_1}\frac{\partial u_1}{\partial x}$$

$$= \frac{\partial u_3}{\partial x} + \sum_{i=1}^{3} \frac{\partial u_3}{\partial u_i}\frac{\partial u_i}{\partial x}$$

Total Derivative Chain Rule

To simplify the formulation of partial derivatives $\frac{\partial y}{\partial x}$

we can say that the "total" partial derivatives is



$$\frac{\partial y}{\partial x} = \frac{\partial f(x,u_1,\ldots,u_n)}{\partial x}$$

$$= \frac{\partial u_3}{\partial x} + \frac{\partial u_3}{\partial u_2}\frac{\partial u_2}{\partial x} + \frac{\partial u_3}{\partial u_1}\frac{\partial u_1}{\partial x}$$

$$= \frac{\partial u_3}{\partial x} + \sum_{i=1}^{2} \frac{\partial u_3}{\partial u_i}\frac{\partial u_i}{\partial x}$$

$$= \frac{\partial u_3}{\partial x} + \sum_{i=1}^{n} \frac{\partial f}{\partial u_i}\frac{\partial u_i}{\partial x}$$

$$= \sum_{i=1}^{n+1} \frac{\partial f}{\partial u_i}\frac{\partial u_i}{\partial x}$$

Matrix Calculus for Deep Learning

Vector chain rule

Vector Chain Rule

Remember ***partial derivatives*** to the $y = f(X)$ equation **w.r.t.** $\boldsymbol{x}$:

$$\frac{\partial y}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial}{\partial x} f_1(X) \\ \frac{\partial}{\partial x} f_2(X) \\ \vdots \\ \frac{\partial}{\partial x} f_m(X) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(X) & \cdots & \frac{\partial}{\partial x_n} f_1(X) \\ \frac{\partial}{\partial x_1} f_2(X) & \cdots & \frac{\partial}{\partial x_n} f_2(X) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_1} f_m(X) & \cdots & \frac{\partial}{\partial x_n} f_{m\partial}(X) \end{bmatrix}$$

Remember ***chain rule,*** if we have $y = f(g(X))$ **:**

$$\frac{\partial y}{\partial \boldsymbol{x}} = \frac{\partial f}{\partial g} \frac{\partial g}{\partial \boldsymbol{x}}$$

Vector Chain Rule

Remember ***chain rule,*** if we have $y = f(g(X))$ **:**

$$\frac{\partial y}{\boldsymbol{\partial x}} = \frac{\partial f}{\partial g}\frac{\partial g}{\boldsymbol{\partial x}}$$

Remember ***partial derivatives*** to the $y = f(g(X))$ equation **w.r.t.** $\boldsymbol{x}$:

$$\frac{\partial y}{\boldsymbol{\partial x}} = \frac{\partial f}{\partial g}\frac{\partial g}{\boldsymbol{\partial x}} = \begin{bmatrix} \frac{\partial}{\partial g_1} f_1(X) & \cdots & \frac{\partial}{\partial g_k} f_1(X) \\ \frac{\partial}{\partial g_1} f_2(X) & \cdots & \frac{\partial}{\partial g_k} f_2(X) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial g_1} f_m(X) & \cdots & \frac{\partial}{\partial g_k} f_m(X) \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x_1} g_1(X) & \cdots & \frac{\partial}{\partial x_n} g_1(X) \\ \frac{\partial}{\partial x_1} g_2(X) & \cdots & \frac{\partial}{\partial x_n} g_2(X) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_1} g_k(X) & \cdots & \frac{\partial}{\partial x_n} g_k(X) \end{bmatrix}$$

Vector Chain Rule

Remember ***partial derivatives*** to the $y = f(g(X))$ equation **w.r.t.** $\boldsymbol{x}$:

$$\frac{\partial y}{\partial \boldsymbol{x}} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial}{\partial g_1} f_1(X) & \cdots & \frac{\partial}{\partial g_k} f_1(X) \\ \frac{\partial}{\partial g_1} f_2(X) & \cdots & \frac{\partial}{\partial g_k} f_2(X) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial g_1} f_m(X) & \cdots & \frac{\partial}{\partial g_k} f_m(X) \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x_1} g_1(X) & \cdots & \frac{\partial}{\partial x_n} g_1(X) \\ \frac{\partial}{\partial x_1} g_2(X) & \cdots & \frac{\partial}{\partial x_n} g_2(X) \\ \vdots & \vdots & \vdots \\ \frac{\partial}{\partial x_1} g_k(X) & \cdots & \frac{\partial}{\partial x_n} g_k(X) \end{bmatrix}$$

where $m = |f|$, $n = |x|$ and $k = |g|$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

Remember this:

$$\frac{\partial y_j}{\partial x} = \sum_{i=1}^{n+1} \frac{\partial f}{\partial u_i} \frac{\partial u_i}{\partial x}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

Remember this:

$$\frac{\partial y_j}{\partial x} = \sum_{i=1}^{n+1} \frac{\partial f}{\partial u_i} \frac{\partial u_i}{\partial x}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

Remember this:

$$\frac{\partial y_j}{\partial x} = \sum_{i=1}^{n+1} \frac{\partial f}{\partial u_i} \frac{\partial u_i}{\partial x}$$

So:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{d}{dx_1} f_1(g_1) \\ \frac{d}{dx_2} f_2(g_2) \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dg_1}\frac{dg_1}{dx_1} + \frac{df_1}{dg_2}\frac{dg_2}{dx_1} \\ \frac{df_2}{dg_1}\frac{dg_1}{dx_1} + \frac{df_2}{dg_2}\frac{dg_2}{dx_1} \end{bmatrix}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

Now we plug in the value:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{d}{dx_1} f_1(g_1) \\ \frac{d}{dx_2} f_2(g_2) \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dg_1}\frac{dg_1}{dx_1} + \frac{df_1}{dg_2}\frac{dg_2}{dx_1} \\ \frac{df_2}{dg_1}\frac{dg_1}{dx_1} + \frac{df_2}{dg_2}\frac{dg_2}{dx_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1} 2x + 0 \\ 0 + \cos(g_2)\, 3 \end{bmatrix}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

Now we plug in the value:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{df_1}{dg_1}\frac{dg_1}{dx_1} + \frac{df_1}{dg_2}\frac{dg_2}{dx_1} \\ \frac{df_2}{dg_1}\frac{dg_1}{dx_1} + \frac{df_2}{dg_2}\frac{dg_2}{dx_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1}2x + 0 \\ 0 + \cos(g_2)\,3 \end{bmatrix} = \begin{bmatrix} \frac{2}{x} \\ 3\cos(3x) \end{bmatrix}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

And if we rearrange the Jacobian:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{d}{dx_1} f_1(g_1) \\ \frac{d}{dx_2} f_2(g_2) \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dg_1}\frac{dg_1}{dx_1} + \frac{df_1}{dg_2}\frac{dg_2}{dx_1} \\ \frac{df_2}{dg_1}\frac{dg_1}{dx_1} + \frac{df_2}{dg_2}\frac{dg_2}{dx_1} \end{bmatrix} =$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ x \end{bmatrix}$$

And if we rearrange the Jacobian:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{d}{dx_1} f_1(g_1) \\ \frac{d}{dx_2} f_2(g_2) \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dg_1}\frac{dg_1}{dx_1} + \frac{df_1}{dg_2}\frac{dg_2}{dx_1} \\ \frac{df_2}{dg_1}\frac{dg_1}{dx_1} + \frac{df_2}{dg_2}\frac{dg_2}{dx_1} \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dg_1} & \frac{df_1}{dg_2} \\ \frac{df_2}{dg_1} & \frac{df_2}{dg_2} \end{bmatrix} \begin{bmatrix} \frac{dg_1}{dx_1} \\ \frac{dg_2}{dx_1} \end{bmatrix}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

And if we rearrange the Jacobian:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{df_1}{dg_1} & \frac{df_1}{dg_2} \\ \frac{df_2}{dg_1} & \frac{df_2}{dg_2} \end{bmatrix} \begin{bmatrix} \frac{dg_1}{dx_1} \\ \frac{dg_2}{dx_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1} & 0 \\ 0 & \cos(g_2) \end{bmatrix} \begin{bmatrix} 2x \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1} 2x + 0 \\ 0 + 3\cos(g_2) \end{bmatrix}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ 3x \end{bmatrix}$$

And if we rearrange the Jacobian:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{df_1}{dg_1} & \frac{df_1}{dg_2} \\ \frac{df_2}{dg_1} & \frac{df_2}{dg_2} \end{bmatrix} \begin{bmatrix} \frac{dg_1}{dx_1} \\ \frac{dg_2}{dx_1} \end{bmatrix} = \begin{bmatrix} \frac{1}{g_1} 2x + 0 \\ 0 + 3\cos(g_2) \end{bmatrix} = \begin{bmatrix} \frac{2}{x} \\ 3\cos(3x) \end{bmatrix}$$

Matrix Calculus for Deep Learning

Summary

Scalar, Partial, Vectorial Derivatives

- Think of ***scalar derivatives*** $\frac{d}{dx}$ as an ***operator that maps a single variable/parameter to a function*** $f(x)$ with respect to (w.r.t.) x

- The ***partial derivatives*** e.g. $[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}]$ ***"individualizes" the operations to map multiple variable/parameter to the multi-variable function*** , e.g. $f(x, y)$

- In vector calculus, we pack ***partial derivatives as a vector*** and ***represent the gradient of a function*** $f(x, y)$ as $\nabla f(x, y) = \left[\frac{\partial}{\partial x} f(x, y), \frac{\partial}{\partial y} f(x, y)\right]$

Scalar, Partial, Vectorial Derivatives

Math	*aka	Math-sy English	Human-ish English
$\frac{dy}{dx}$	$f'(x)$	Scalar derivatives	Calculates changes for one variable
	$d_x f$		
	$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$		
$\frac{\partial y}{\partial x}$	$f'(x)$	Partial derivatives	Calculates changes for multiple variable in parts
	$\partial_x f$		
	$\lim_{h \to 0} \frac{f(x_i, \ldots, x_i + h, \ldots, x_n) - f(x_i, \ldots, x_n)}{h}$		
$\nabla f(x, y)$		Vectorized gradients	Calculates changes for multiple variables in parts in a vectorized manner

Jacobian

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x} f_1(\boldsymbol{X}) \\ \frac{\partial}{\partial x} f_2(\boldsymbol{X}) \\ \vdots \\ \frac{\partial}{\partial x} f_m(\boldsymbol{X}) \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(\boldsymbol{X}) & \frac{\partial}{\partial x_2} f_1(\boldsymbol{X}) & \dots & \frac{\partial}{\partial x_n} f_1(\boldsymbol{X}) \\ \frac{\partial}{\partial x_1} f_2(\boldsymbol{X}) & \frac{\partial}{\partial x_2} f_2(\boldsymbol{X}) & \dots & \frac{\partial}{\partial x_n} f_2(\boldsymbol{X}) \\ & & \vdots & \\ \frac{\partial}{\partial x_1} f_m(\boldsymbol{X}) & \frac{\partial}{\partial x_2} f_m(\boldsymbol{X}) & \dots & \frac{\partial}{\partial x_n} f_m(\boldsymbol{X}) \end{bmatrix}$$

Derivatives of Vectors with Element-wise Binary Operators

Similarly **w.r.t. x** :

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1(W) \bigcirc g_1(X) & 0 & 0 & 0 \\ 0 & \frac{\partial}{\partial x_2} f_2(W) \bigcirc g_2(X) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \frac{\partial}{\partial x_n} f_n(W) \bigcirc g_n(X) \end{bmatrix}$$

We could rewrite it as such:

$$\frac{\partial y}{\partial x} = diag(\frac{\partial}{\partial x_1} f_1(W) \bigcirc g_1(X), \frac{\partial}{\partial x_2} f_2(W) \bigcirc g_2(X), \ldots, \frac{\partial}{\partial x_n} f_n(W) \bigcirc g_n(X))$$

Special Cases for Jacobian Partial Derivatives

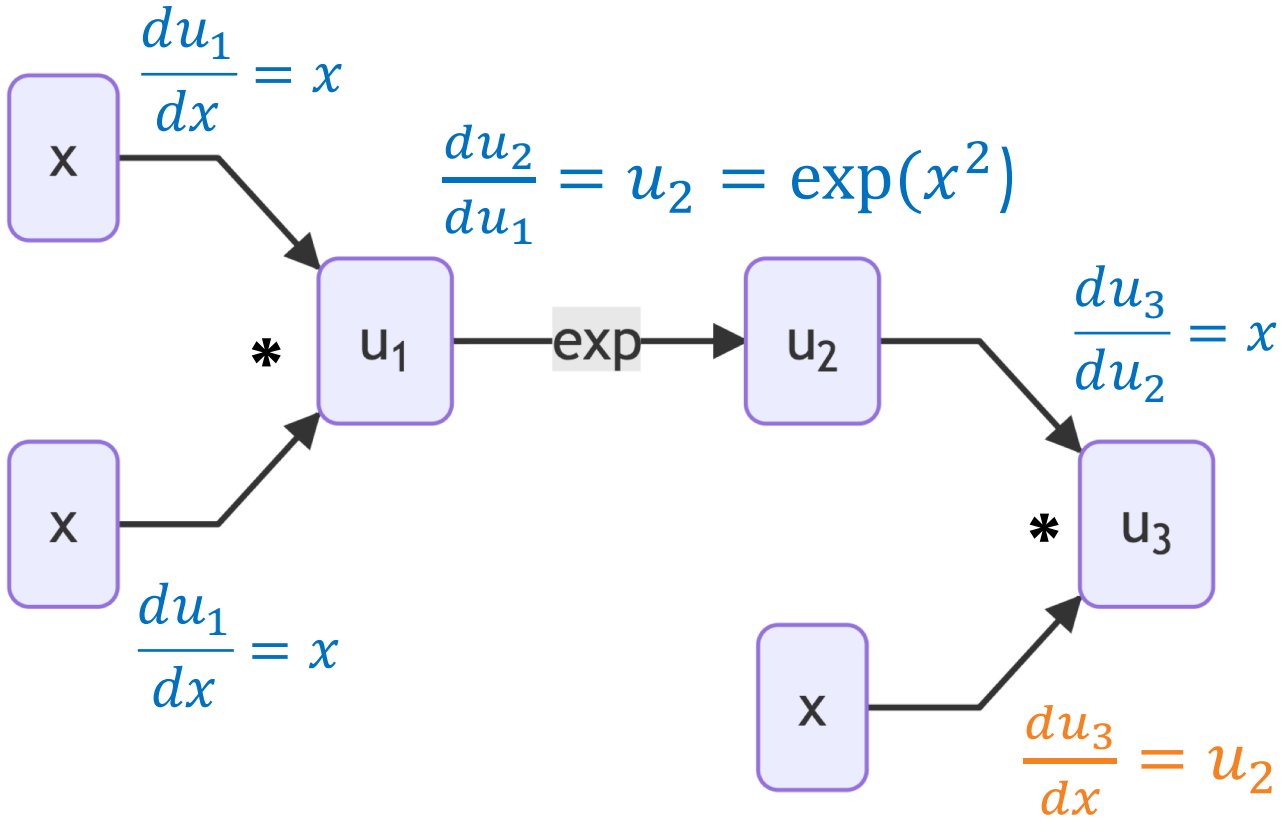
Op	Partial with respect to $\mathbf{w}$
$+$	$\frac{\partial(\mathbf{w}+\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i+x_i)}{\partial w_i} \ldots) = diag(\vec{1}) = I$
$-$	$\frac{\partial(\mathbf{w}-\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i-x_i)}{\partial w_i} \ldots) = diag(\vec{1}) = I$
$\otimes$	$\frac{\partial(\mathbf{w}\otimes\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i\times x_i)}{\partial w_i} \ldots) = diag(\mathbf{x})$
$\oslash$	$\frac{\partial(\mathbf{w}\oslash\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i/x_i)}{\partial w_i} \ldots) = diag(\ldots \frac{1}{x_i} \ldots)$

Op	Partial with respect to $\mathbf{x}$
$+$	$\frac{\partial(\mathbf{w}+\mathbf{x})}{\partial \mathbf{x}} = I$
$-$	$\frac{\partial(\mathbf{w}-\mathbf{x})}{\partial \mathbf{x}} = diag(\ldots \frac{\partial(w_i-x_i)}{\partial x_i} \ldots) = diag(-\vec{1}) = -I$
$\otimes$	$\frac{\partial(\mathbf{w}\otimes\mathbf{x})}{\partial \mathbf{x}} = diag(\mathbf{w})$
$\oslash$	$\frac{\partial(\mathbf{w}\oslash\mathbf{x})}{\partial \mathbf{x}} = diag(\ldots \frac{-w_i}{x_i^2} \ldots)$

(Parr and Howard, 2018)

Partial Derivatives thru Computation Graph

$$
\begin{aligned}
 & u_1 = f_1(x) = x^2 \\
 & u_2 = f_2(x) = \exp(u_1) \\
y = & u_3 = f_3(x) = xu_2
\end{aligned}
$$



To compute the partial derivative of $\frac{\partial y}{\partial x}$

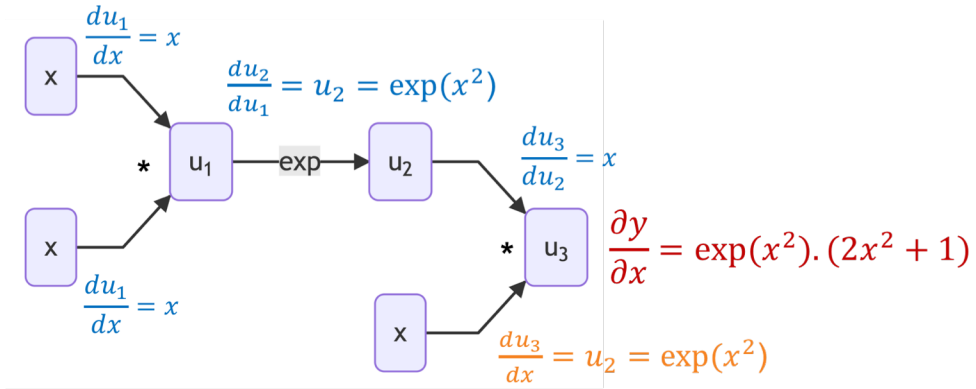
go backwards from the last node to every x node, multiply all intermediate the derivatives on the edges and sum them

$$
\begin{aligned}
\frac{\partial y}{\partial x} = & x.\exp(x^2).x \\
& + x.\exp(x^2).x \\
& + \exp(x^2)
\end{aligned}
$$

Total Derivative Chain Rule

To simplify the formulation of partial derivatives $\frac{\partial y}{\partial x}$

we can say that the "total" partial derivatives is



$$\frac{\partial y}{\partial x} = \frac{\partial f(x,u_1,\ldots,u_n)}{\partial x}$$

$$= \frac{\partial u_3}{\partial x} + \frac{\partial u_3}{\partial u_2}\frac{\partial u_2}{\partial x} + \frac{\partial u_3}{\partial u_1}\frac{\partial u_1}{\partial x}$$

$$= \frac{\partial u_3}{\partial x} + \sum_{i=1}^{3} \frac{\partial u_3}{\partial u_i}\frac{\partial u_i}{\partial x}$$

$$= \frac{\partial u_3}{\partial x} + \sum_{i=1}^{n} \frac{\partial f}{\partial u_i}\frac{\partial u_i}{\partial x}$$

$$= \sum_{i=1}^{n+1} \frac{\partial f}{\partial u_i}\frac{\partial u_i}{\partial x}$$

Vector Chain Rule

Consider y, f, g as vectors in $y = f(g(X))$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} f_1(g_1(x)) \\ f_2(g_2(x)) \end{bmatrix} \quad \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} \ln(g_1) \\ \sin(g_2) \end{bmatrix} \quad \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} x^2 \\ x \end{bmatrix}$$

And if we rearrange the Jacobian:

$$\frac{\partial y}{\partial x} = \begin{bmatrix} \frac{d}{dx_1} f_1(g_1) \\ \frac{d}{dx_2} f_2(g_2) \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dg_1}\frac{dg_1}{dx_1} + \frac{df_1}{dg_2}\frac{dg_2}{dx_1} \\ \frac{df_2}{dg_1}\frac{dg_1}{dx_1} + \frac{df_2}{dg_2}\frac{dg_2}{dx_1} \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dg_1} & \frac{df_1}{dg_2} \\ \frac{df_2}{dg_1} & \frac{df_2}{dg_2} \end{bmatrix} \begin{bmatrix} \frac{dg_1}{dx_1} \\ \frac{dg_2}{dx_1} \end{bmatrix}$$

Backpropagation

Jacobian refresher

Special Cases for Jacobian Partial Derivatives

Op	Partial with respect to $\mathbf{w}$
$+$	$\frac{\partial(\mathbf{w}+\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i+x_i)}{\partial w_i} \ldots) = diag(\vec{1}) = I$
$-$	$\frac{\partial(\mathbf{w}-\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i-x_i)}{\partial w_i} \ldots) = diag(\vec{1}) = I$
$\otimes$	$\frac{\partial(\mathbf{w}\otimes\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i\times x_i)}{\partial w_i} \ldots) = diag(\mathbf{x})$
$\oslash$	$\frac{\partial(\mathbf{w}\oslash\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i/x_i)}{\partial w_i} \ldots) = diag(\ldots \frac{1}{x_i} \ldots)$

Op	Partial with respect to $\mathbf{x}$
$+$	$\frac{\partial(\mathbf{w}+\mathbf{x})}{\partial \mathbf{x}} = I$
$-$	$\frac{\partial(\mathbf{w}-\mathbf{x})}{\partial \mathbf{x}} = diag(\ldots \frac{\partial(w_i-x_i)}{\partial x_i} \ldots) = diag(-\vec{1}) = -I$
$\otimes$	$\frac{\partial(\mathbf{w}\otimes\mathbf{x})}{\partial \mathbf{x}} = diag(\mathbf{w})$
$\oslash$	$\frac{\partial(\mathbf{w}\oslash\mathbf{x})}{\partial \mathbf{x}} = diag(\ldots \frac{-w_i}{x_i^2} \ldots)$

(Parr and Howard, 2018)

Jacobian Matrix: Generalization of the Gradient

Given a function ***f(x)*** with ***m*** **outputs** and ***n*** **inputs**:

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(x_1, x_2, \ldots, x_n), \ldots, f_m(x_1, x_2, \ldots, x_n)]$$

Jacobian:

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Inside the Jacobian:

$$\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right)_{ij} = \frac{\partial f_i}{\partial x_j}$$

Jacobian of an Activation Function

$$y = f(u) \qquad \text{\# Activation}$$
$$u = Wx + b \qquad \text{\# Affine function}$$

Partial derivatives of activation:

$$\left(\frac{\partial y}{\partial u}\right)_{ij} = \frac{\partial y_i}{\partial u_j} = \frac{\partial}{\partial u_j} f(u_i)$$

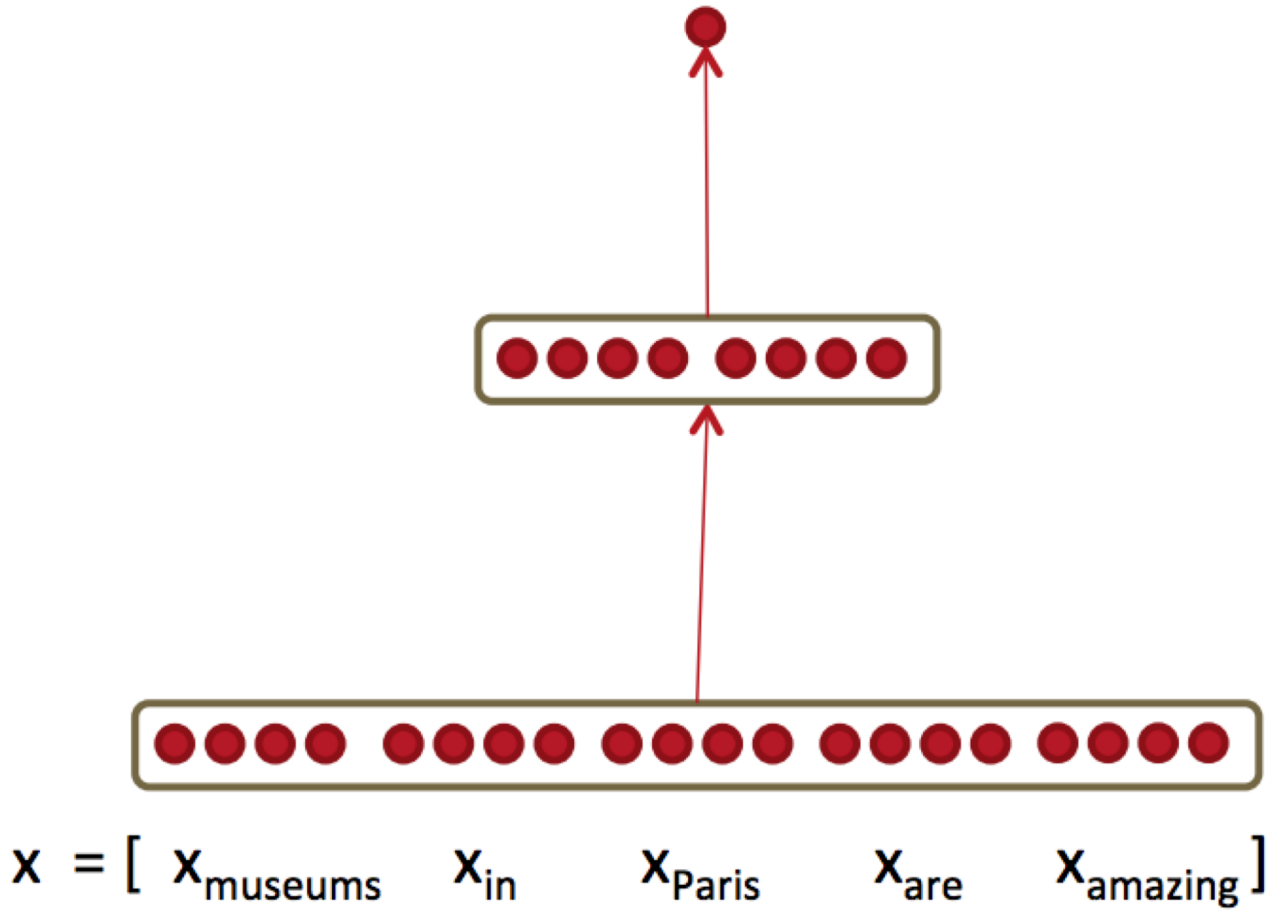
$$= \begin{pmatrix} \frac{\partial}{\partial u_1} f(u_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \frac{\partial}{\partial u_n} f(u_n) \end{pmatrix} = \text{diag}\left(\frac{\partial}{\partial u} f(u)\right)$$

Neural Nets

$$s = \boldsymbol{u}^T \boldsymbol{h}$$

$$\boldsymbol{h} = f(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b})$$

$$\boldsymbol{x} \quad \text{(input)}$$



Other Useful Jacobian

$$\frac{\partial}{\partial \boldsymbol{x}}(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}) = \boldsymbol{W}$$

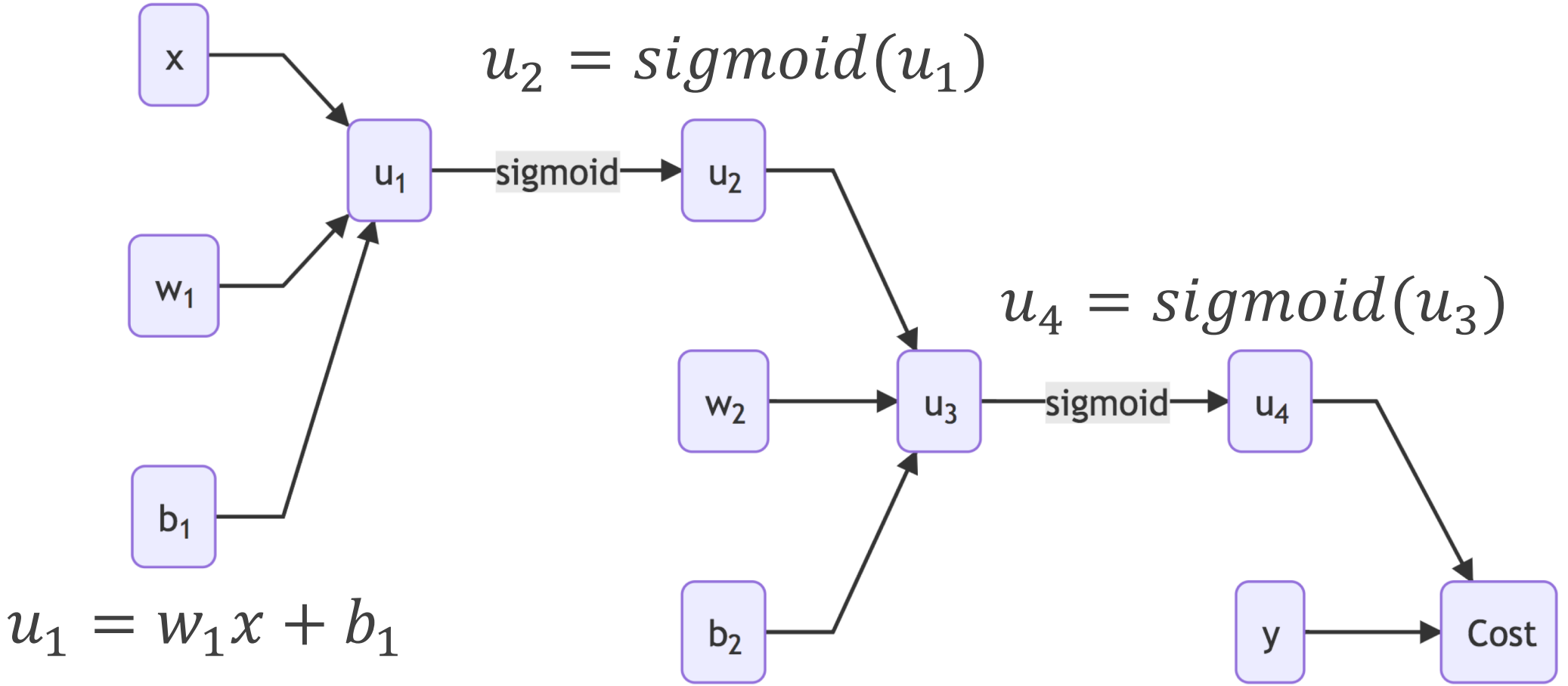
$$\frac{\partial}{\partial \boldsymbol{b}}(\boldsymbol{W}\boldsymbol{x}+\boldsymbol{b}) = \boldsymbol{I} \quad (\text{Identity matrix})$$

$$\frac{\partial}{\partial \boldsymbol{u}}(\boldsymbol{u}^T\boldsymbol{h}) = \boldsymbol{h}^{\boldsymbol{T}}$$

Backpropagation

For real this time…

Derivatives of a Multi-Layered Perceptron



$u_2 = sigmoid(u_1)$

$u_4 = sigmoid(u_3)$

$u_1 = w_1 x + b_1$

$u_3 = w_2 u_2 + b_2$

$cost = L(y, u_3)$

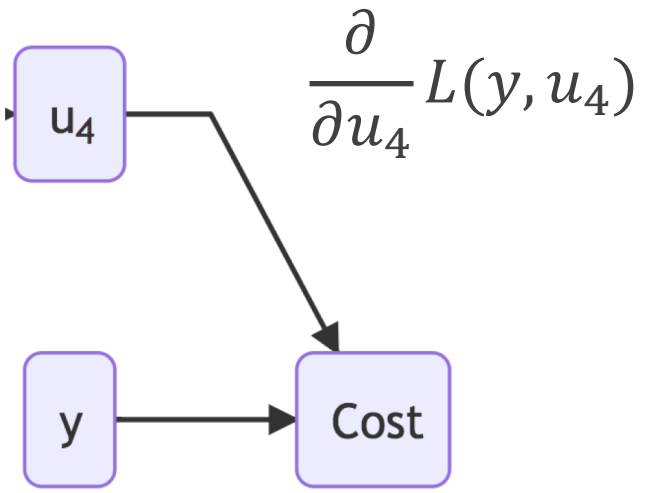
Derivatives of a Multi-Layered Perceptron

u_4 is our predictions from the MLP net

y is the truth from the training data

Assume we're doing classification and use Cross Entropy (CE) as our loss function:

$$\begin{bmatrix} u_{4_1} \\ \vdots \\ u_{4_k} \\ \vdots \\ u_{4_m} \end{bmatrix} \longrightarrow \textbf{Cost} \longleftarrow \begin{bmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_m \end{bmatrix} \qquad \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}$$



$$cost = L(y, u_4)$$

Derivatives of a Multi-Layered Perceptron

u_4 is our predictions from the MLP net

y is the truth from the training data

Assume we're doing classification and Cross Entropy (CE) as our loss function:

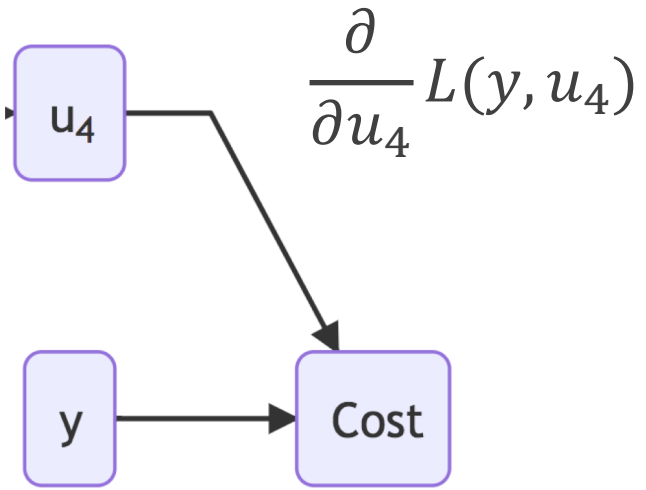
$$L(y, u_4) = C(y, u_4) = [c_1 \quad \dots \quad c_m]$$

$$c_i = -log(u_{4_i})$$

$$\frac{\partial}{\partial u_4} c_i = \frac{-1}{u_{4_i}}$$

$$\frac{\partial}{\partial u_4} L(y, u_4) = [c_1 \quad \dots \quad c_k \quad \dots \quad c_m]$$

$$= \left[0 \quad \dots \quad \frac{-1}{u_{4_k}} \quad \dots \quad 0\right]$$



$$\frac{\partial}{\partial u_4} L(y, u_4)$$

$$cost = L(y, u_4)$$

Derivatives of a Multi-Layered Perceptron

Continue backwards…

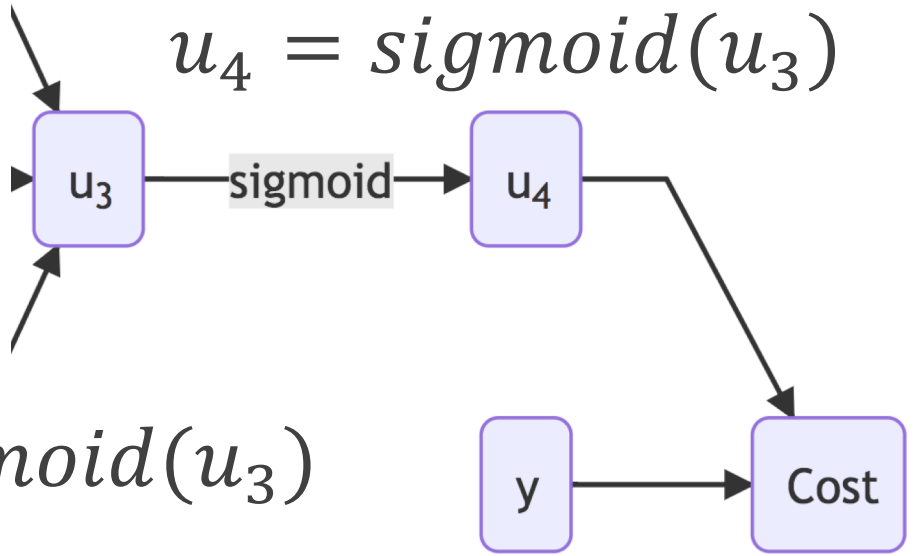
$$\frac{\partial u_4}{\partial u_3} = ???$$

$$\left(\frac{\partial u_4}{\partial u_3}\right)_{ij} = \frac{\partial u_{4_i}}{\partial u_{3_j}}$$

When $i \neq j$, $\frac{\partial u_{4_i}}{\partial u_{3_j}} = 0$

When $i = j$, $\frac{\partial u_{4_i}}{\partial u_{3_j}} = \frac{\partial}{\partial u_3} sigmoid(u_3)$

$$u_4 = sigmoid(u_3)$$



$$cost = L(y, u_4)$$

Derivatives of a Multi-Layered Perceptron

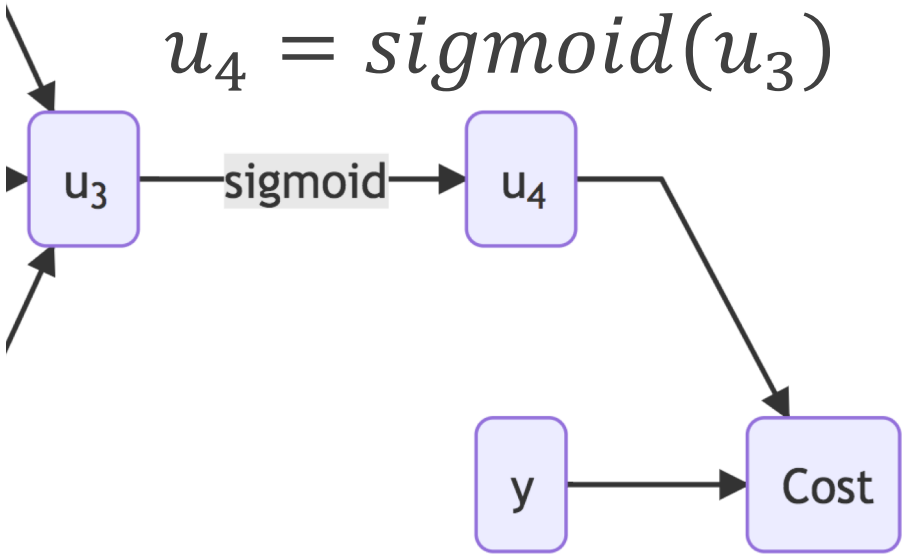
Continue backwards…

$$\frac{\partial u_4}{\partial u_3} = ???$$

$$\left(\frac{\partial u_4}{\partial u_3}\right)_{ij} = \frac{\partial u_{4_i}}{\partial u_{3_j}}$$

When $i \neq j$, $\frac{\partial u_{4_i}}{\partial u_{3_j}} = 0$

$$u_4 = sigmoid(u_3)$$



$$cost = L(y, u_4)$$

Derivatives of a Multi-Layered Perceptron

Continue backwards…

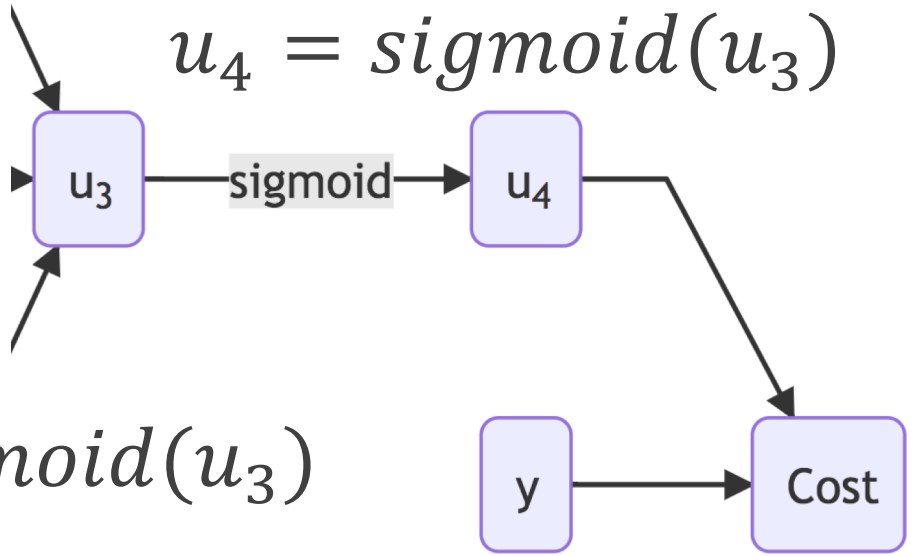
$$\frac{\partial u_4}{\partial u_3} = ???$$

$$\left(\frac{\partial u_4}{\partial u_3}\right)_{ij} = \frac{\partial u_{4_i}}{\partial u_{3_j}}$$

When $i \neq j$, $\frac{\partial u_{4_i}}{\partial u_{3_j}} = 0$

When $i = j$, $\frac{\partial u_{4_i}}{\partial u_{3_j}} = \frac{\partial}{\partial u_3} sigmoid(u_3)$

$$u_4 = sigmoid(u_3)$$



$$cost = L(y, u_4)$$

Derivatives of a Multi-Layered Perceptron

Continue backwards…

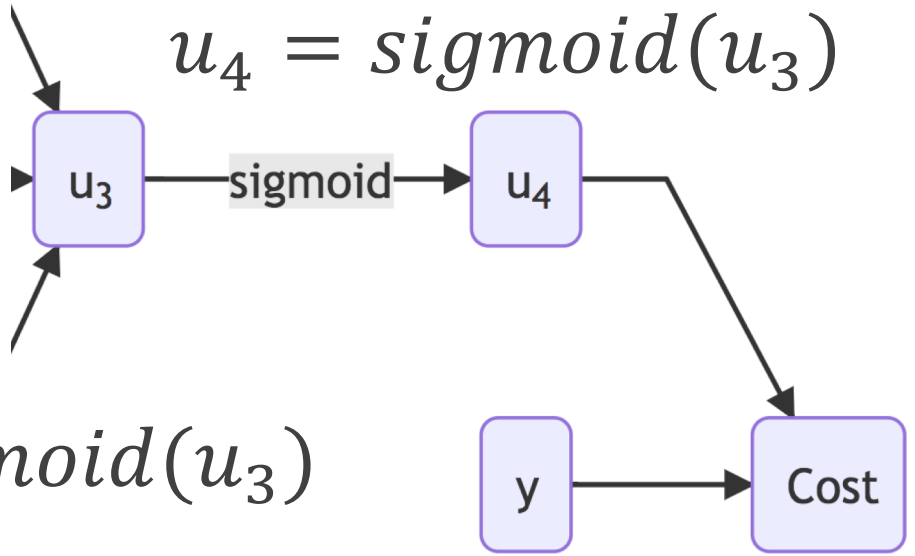
$$\frac{\partial u_4}{\partial u_3} = diag(\frac{\partial u_4}{\partial u_3})$$

$$\left(\frac{\partial u_4}{\partial u_3}\right)_{ij} = \frac{\partial u_{4_i}}{\partial u_{3_j}}$$

When $i \neq j$, $\frac{\partial u_{4_i}}{\partial u_{3_j}} = 0$

When $i = j$, $\frac{\partial u_{4_i}}{\partial u_{3_j}} = \frac{\partial}{\partial u_3} sigmoid(u_3)$

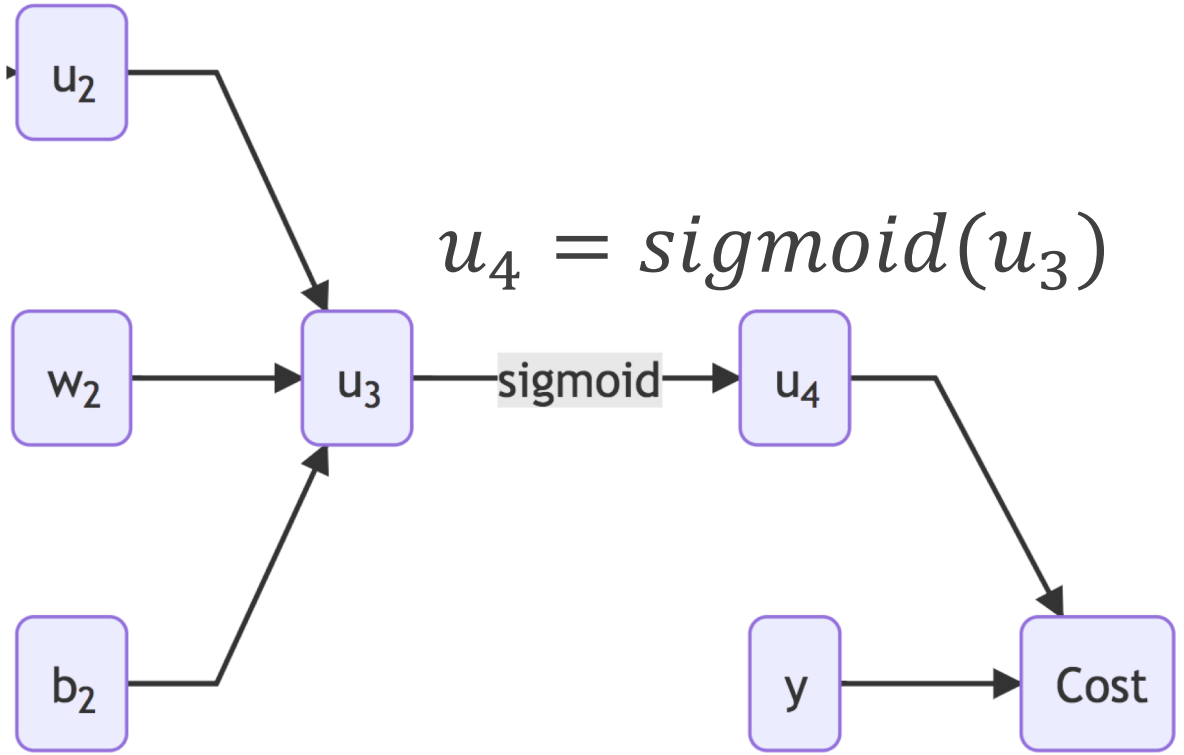
$$u_4 = sigmoid(u_3)$$



$$cost = L(y, u_4)$$

Derivatives of a Multi-Layered Perceptron

$$u_2 = sigmoid(u_1)$$



$$u_4 = sigmoid(u_3)$$

$$u_3 = w_2 u_2 + b_2$$

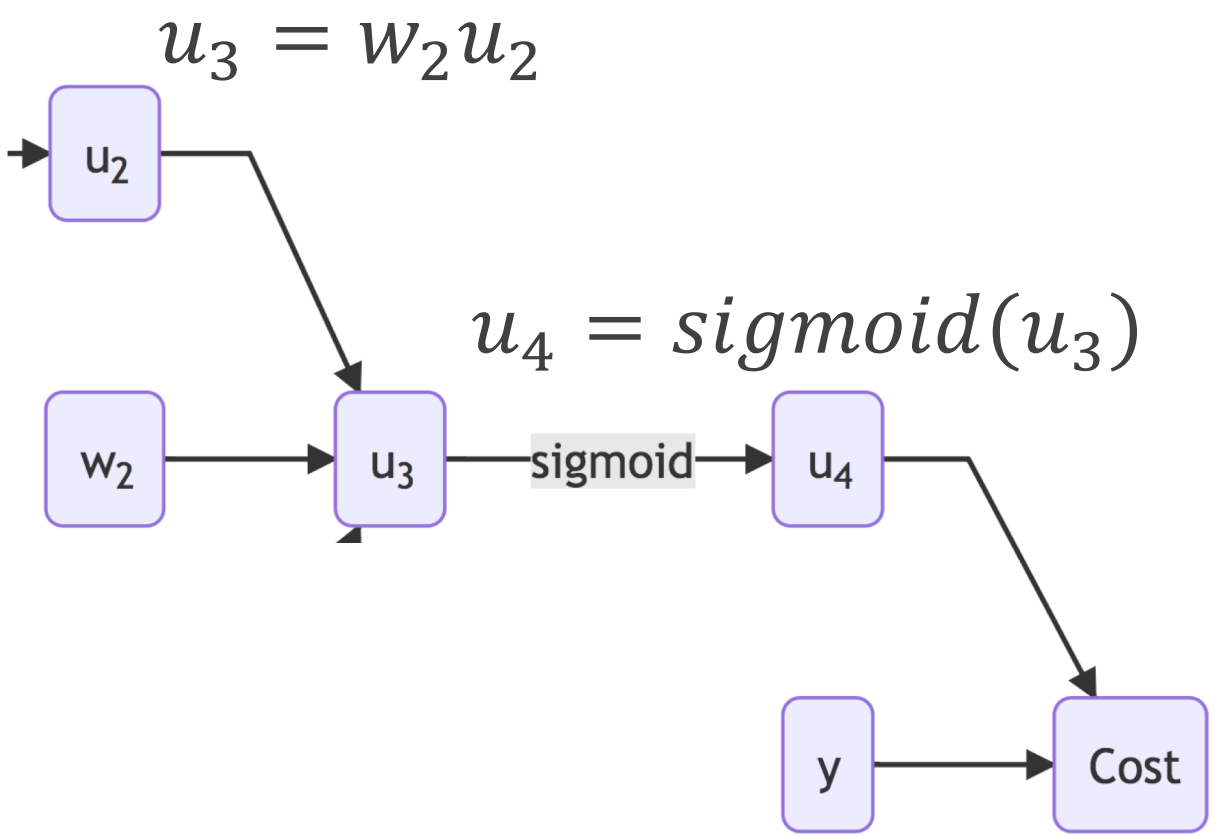
$$cost = L(y, u_4)$$

Derivatives of a Multi-Layered Perceptron

Continue backwards…

$$\frac{\partial u_3}{\partial u_2} = ???$$

$$\frac{\partial u_3}{\partial w_2} = ???$$



$$cost = L(y, u_4)$$

Special Cases for Jacobian Partial Derivatives

Op	Partial with respect to $\mathbf{w}$
$+$	$\frac{\partial(\mathbf{w}+\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i+x_i)}{\partial w_i} \ldots) = diag(\vec{1}) = I$
$-$	$\frac{\partial(\mathbf{w}-\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i-x_i)}{\partial w_i} \ldots) = diag(\vec{1}) = I$
$\otimes$	$\frac{\partial(\mathbf{w}\otimes\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i\times x_i)}{\partial w_i} \ldots) = diag(\mathbf{x})$
$\oslash$	$\frac{\partial(\mathbf{w}\oslash\mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i/x_i)}{\partial w_i} \ldots) = diag(\ldots \frac{1}{x_i} \ldots)$

Op	Partial with respect to $\mathbf{x}$
$+$	$\frac{\partial(\mathbf{w}+\mathbf{x})}{\partial \mathbf{x}} = I$
$-$	$\frac{\partial(\mathbf{w}-\mathbf{x})}{\partial \mathbf{x}} = diag(\ldots \frac{\partial(w_i-x_i)}{\partial x_i} \ldots) = diag(-\vec{1}) = -I$
$\otimes$	$\frac{\partial(\mathbf{w}\otimes\mathbf{x})}{\partial \mathbf{x}} = diag(\mathbf{w})$
$\oslash$	$\frac{\partial(\mathbf{w}\oslash\mathbf{x})}{\partial \mathbf{x}} = diag(\ldots \frac{-w_i}{x_i^2} \ldots)$

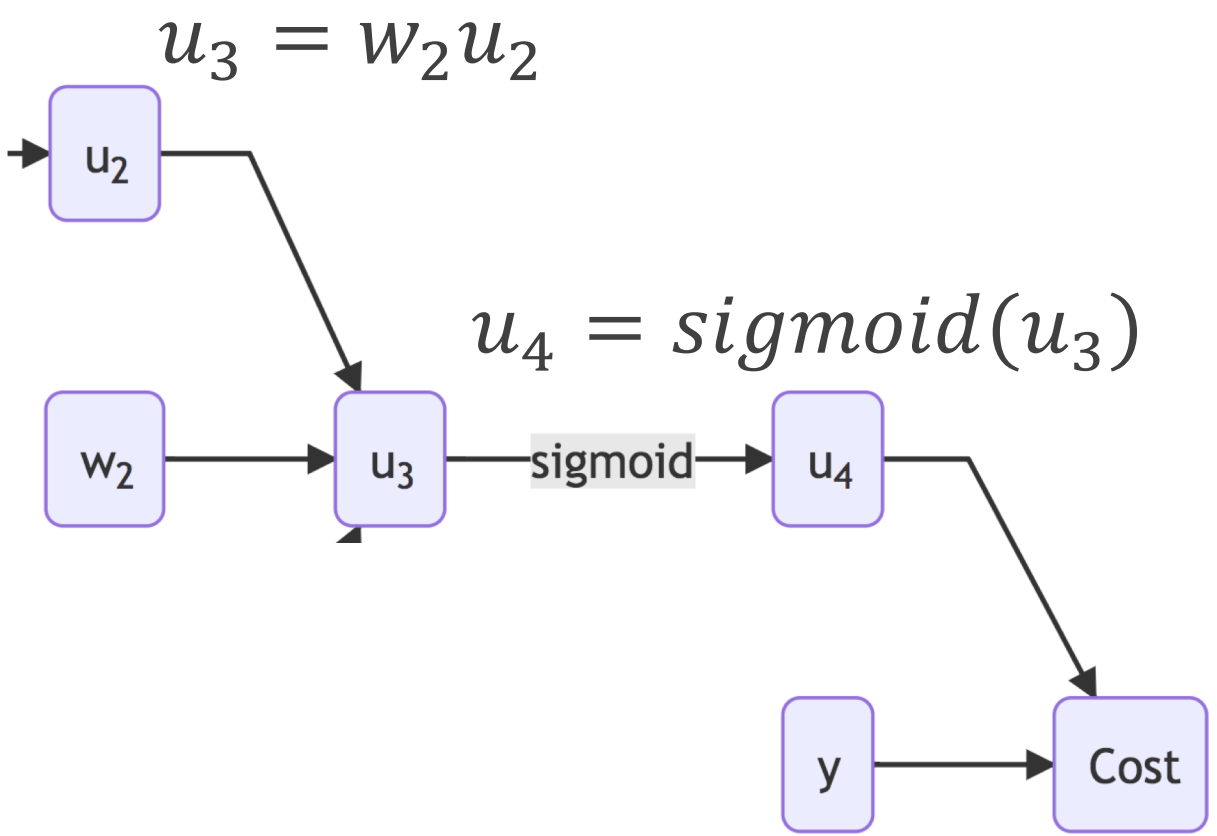
(Parr and Howard, 2018)

Derivatives of a Multi-Layered Perceptron

Continue backwards…

$$\frac{\partial u_3}{\partial u_2} = diag(w_2)$$

$$\frac{\partial u_3}{\partial w_2} = \text{diag}(u_2)$$



Useful Jacobian:

$$\frac{\partial(\mathbf{w} \otimes \mathbf{x})}{\partial \mathbf{x}} = diag(\mathbf{w})$$

$$\frac{\partial(\mathbf{w} \otimes \mathbf{x})}{\partial \mathbf{w}} = diag(\ldots \frac{\partial(w_i \times x_i)}{\partial w_i} \ldots) = diag(\mathbf{x})$$

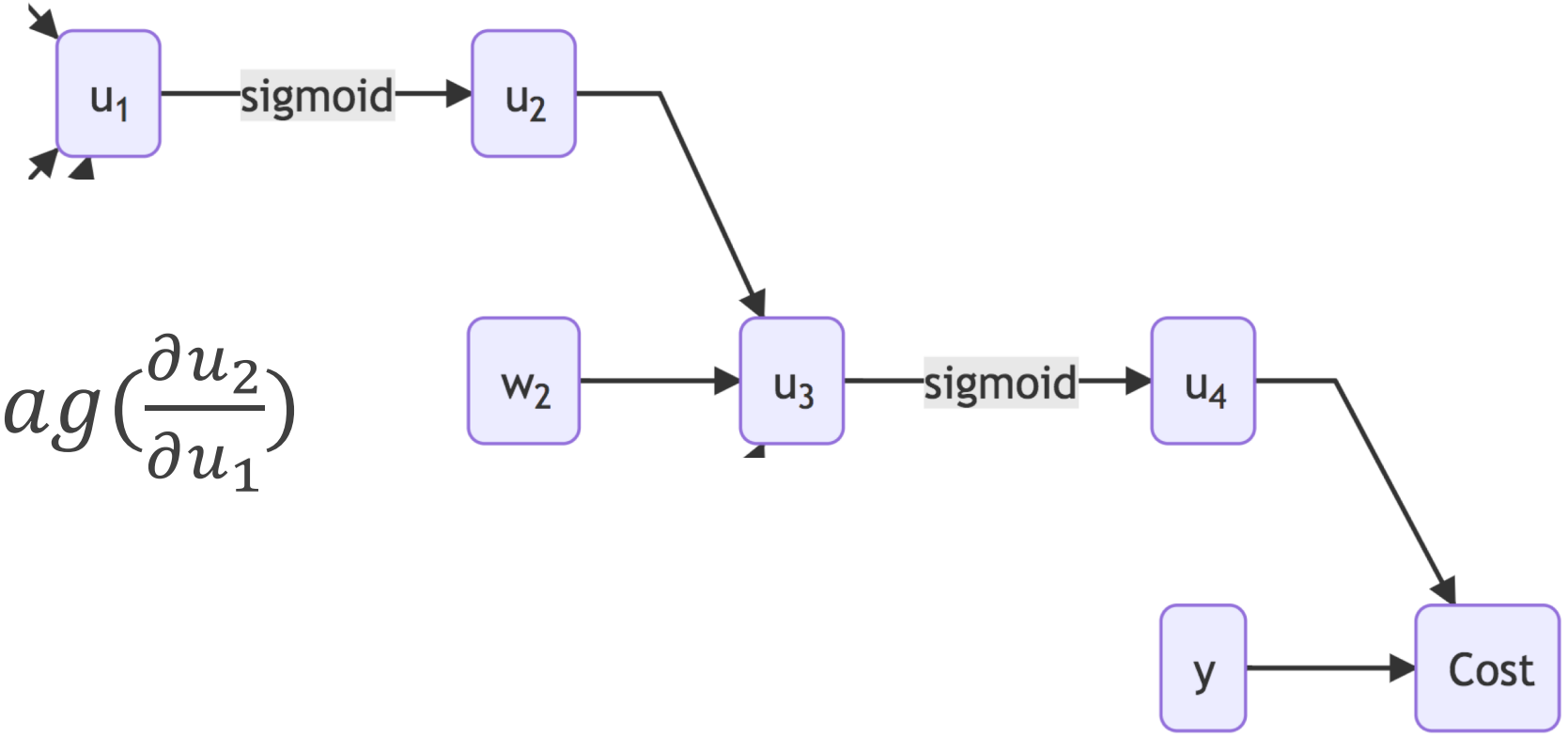
$$cost = L(y, u_4)$$

Derivatives of a Multi-Layered Perceptron

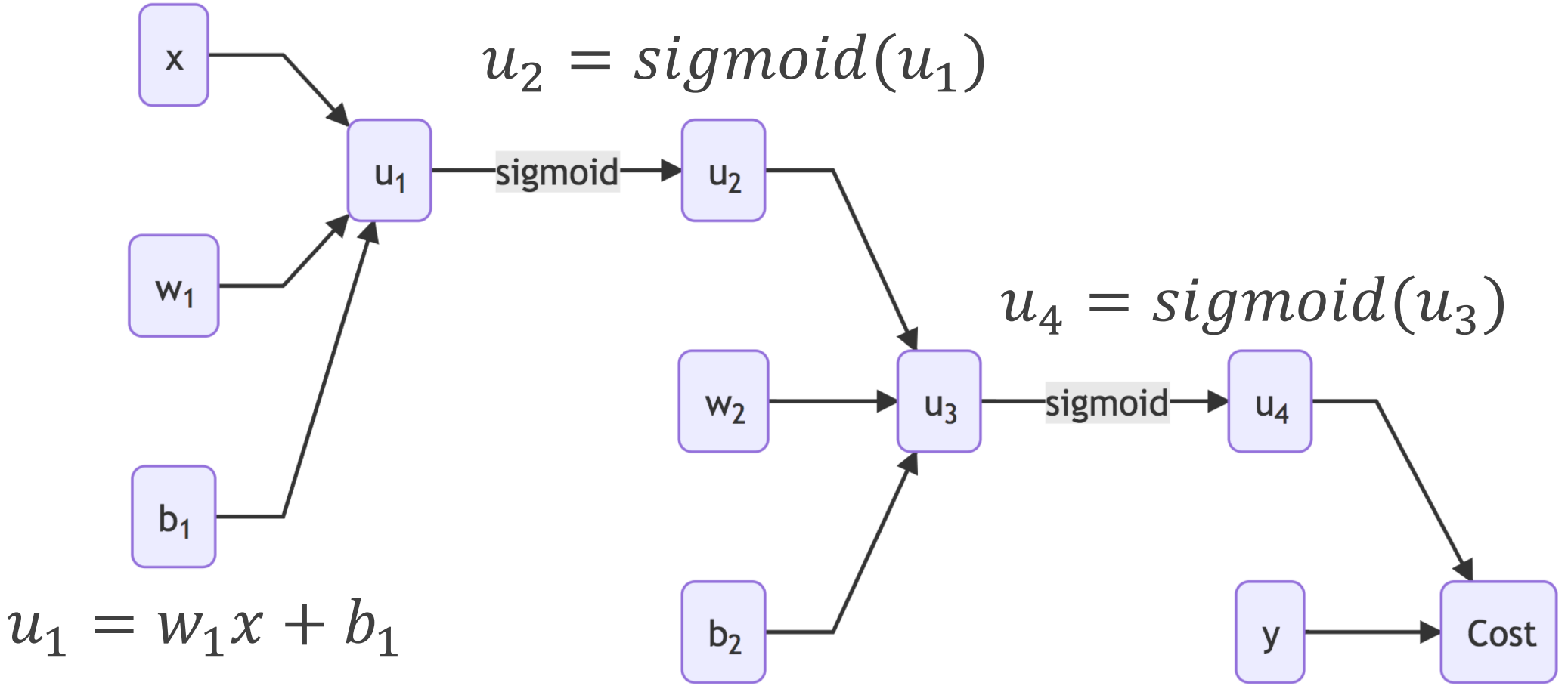
Continue backwards…

$$u_2 = sigmoid(u_1)$$

$$\frac{\partial u_2}{\partial u_1} = diag\left(\frac{\partial u_2}{\partial u_1}\right)$$



Derivatives of a Multi-Layered Perceptron



$u_2 = sigmoid(u_1)$

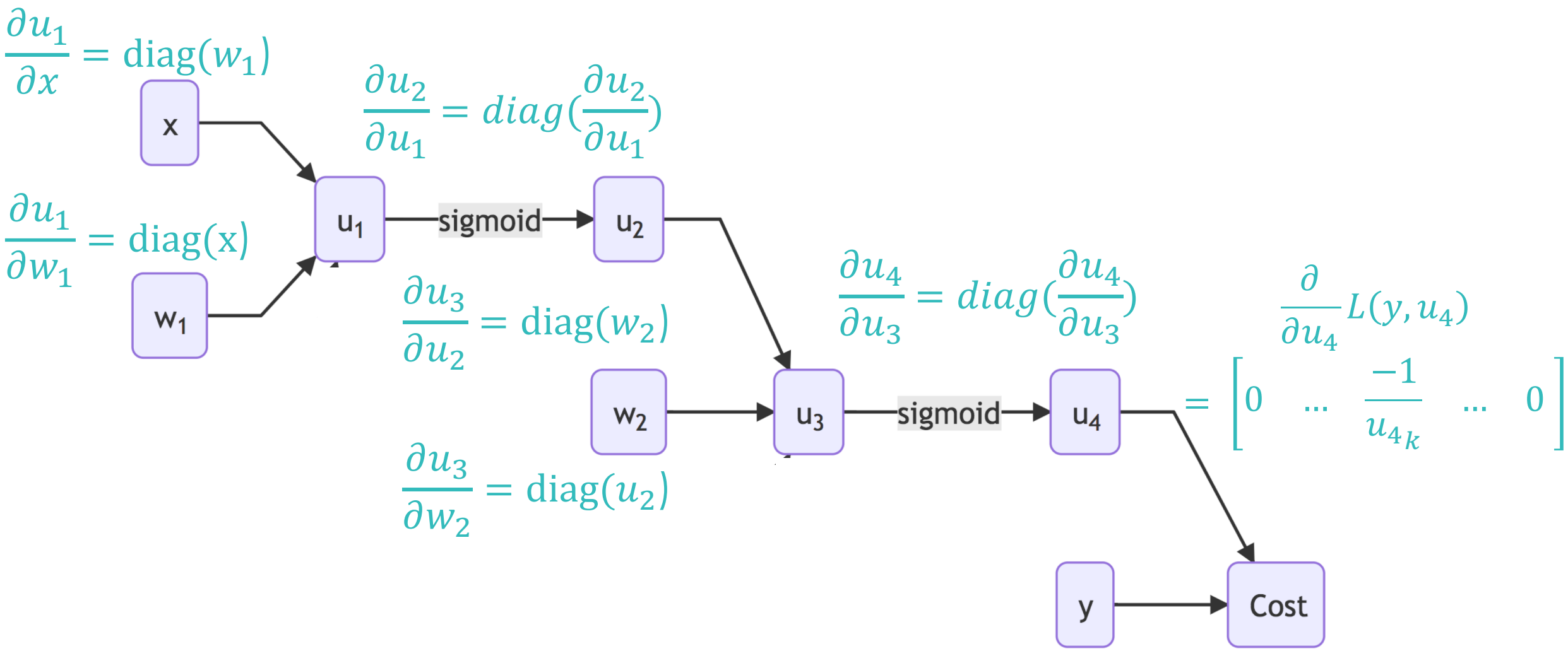
$u_4 = sigmoid(u_3)$

$u_1 = w_1 x + b_1$

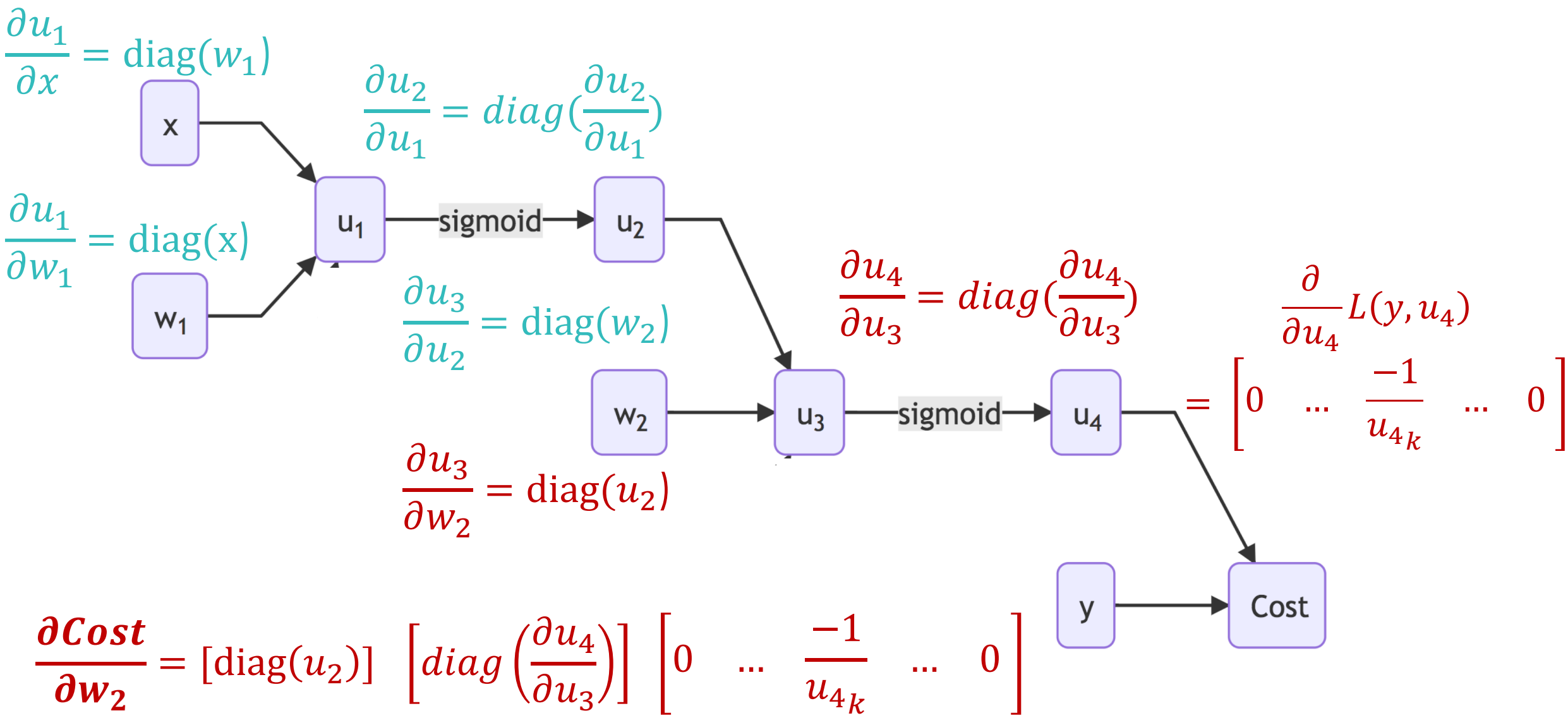
$u_3 = w_2 u_2 + b_2$

$cost = L(y, u_4)$

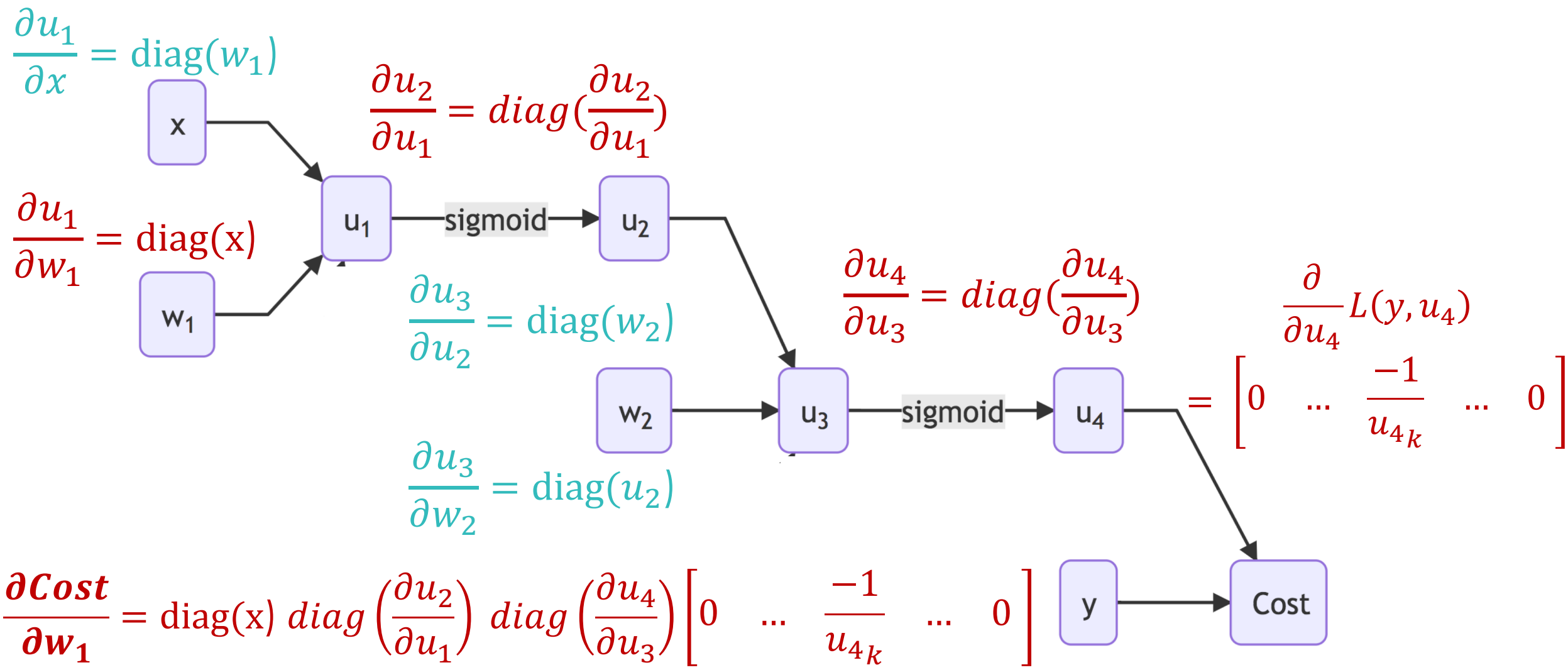
Derivatives of a Multi-Layered Perceptron



Derivatives of a Multi-Layered Perceptron



Derivatives of a Multi-Layered Perceptron



Derivatives of a Multi-Layered Perceptron

$$\frac{\partial Cost}{\partial w_1} = \text{diag(x)}\ diag\left(\frac{\partial u_2}{\partial u_1}\right)\ diag\left(\frac{\partial u_4}{\partial u_3}\right)\begin{bmatrix} 0 & \dots & \frac{-1}{u_{4_k}} & \dots & 0 \end{bmatrix}$$

$$\frac{\partial Cost}{\partial w_2} = \text{diag}(u_2)\ diag\left(\frac{\partial u_4}{\partial u_3}\right)\begin{bmatrix} 0 & \dots & \frac{-1}{u_{4_k}} & \dots & 0 \end{bmatrix}$$

$$w_2 \mathrel{+}= -lr * \frac{\partial Cost}{\partial w_2}$$

$$w_1 \mathrel{+}= -lr * \frac{\partial Cost}{\partial w_1}$$

Fin

We now have all of the pieces needed to compute the derivative of a typical neuron activation for a single neural network computation unit with respect to the model parameters, $\mathbf{w}$ and b:

$$activation(\mathbf{x}) = max(0, \mathbf{w} \cdot \mathbf{x} + b)$$

(This represents a neuron with fully connected weights and rectified linear unit activation. There are, however, other affine functions such as convolution and other activation functions, such as exponential linear units, that follow similar logic.)

Let's worry about *max* later and focus on computing $\frac{\partial}{\partial \mathbf{w}}(\mathbf{w} \cdot \mathbf{x} + b)$ and $\frac{\partial}{\partial b}(\mathbf{w} \cdot \mathbf{x} + b)$. (Recall that neural networks learn through optimization of their weights and biases.) We haven't discussed the derivative of the dot product yet, $y = \mathbf{f}(\mathbf{w}) \cdot \mathbf{g}(\mathbf{x})$, but we can use the chain rule to avoid having to memorize yet another rule. (Note notation y not $\mathbf{y}$ as the result is a scalar not a vector.)

The dot product $\mathbf{w} \cdot \mathbf{x}$ is just the summation of the element-wise multiplication of the elements: $\sum_i^n (w_i x_i) = sum(\mathbf{w} \otimes \mathbf{x})$. (You might also find it useful to remember the linear algebra notation $\mathbf{w} \cdot \mathbf{x} = \mathbf{w}^T \mathbf{x}$.) We know how to compute the partial derivatives of $sum(\mathbf{x})$ and $\mathbf{w} \otimes \mathbf{x}$ but haven't looked at partial derivatives for $sum(\mathbf{w} \otimes \mathbf{x})$. We need the chain rule for that and so we can introduce an intermediate vector variable $\mathbf{u}$ just as we did using the single-variable chain rule:

$$\begin{aligned} \mathbf{u} &= \mathbf{w} \otimes \mathbf{x} \\ y &= sum(\mathbf{u}) \end{aligned}$$

Once we've rephrased y, we recognize two subexpressions for which we already know the partial derivatives:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial \mathbf{w}} &= \frac{\partial}{\partial \mathbf{w}}(\mathbf{w} \otimes \mathbf{x}) &= diag(\mathbf{x}) \\ \frac{\partial y}{\partial \mathbf{u}} &= \frac{\partial}{\partial \mathbf{u}} sum(\mathbf{u}) &= \vec{1}^T \end{aligned}$$

The vector chain rule says to multiply the partials:

$$\frac{\partial y}{\partial \mathbf{w}} = \frac{\partial y}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{w}} = \vec{1}^T diag(\mathbf{x}) = \mathbf{x}^T$$

To check our results, we can grind the dot product down into a pure scalar function:

$$\begin{array}{lllllll} y & = & \mathbf{w} \cdot \mathbf{x} & = & \sum_i^n (w_i x_i) & & \\ \frac{\partial y}{\partial w_j} & = & \frac{\partial}{\partial w_j} \sum_i (w_i x_i) & = & \sum_i \frac{\partial}{\partial w_j} (w_i x_i) & = & \frac{\partial}{\partial w_j} (w_j x_j) & = & x_j \end{array}$$

Then:

$$\frac{\partial y}{\partial \mathbf{w}} = [x_1, \ldots, x_n] = \mathbf{x}^T$$

Hooray! Our scalar results match the vector chain rule results.

Now, let $y = \mathbf{w} \cdot \mathbf{x} + b$, the full expression within the *max* activation function call. We have two different partials to compute, but we don't need the chain rule:

$$\begin{array}{lllll} \frac{\partial y}{\partial \mathbf{w}} & = & \frac{\partial}{\partial \mathbf{w}} \mathbf{w} \cdot \mathbf{x} + \frac{\partial}{\partial \mathbf{w}} b & = & \mathbf{x}^T + \vec{0}^T & = & \mathbf{x}^T \\ \frac{\partial y}{\partial b} & = & \frac{\partial}{\partial b} \mathbf{w} \cdot \mathbf{x} + \frac{\partial}{\partial b} b & = & 0 + 1 & = & 1 \end{array}$$

Let's tackle the partials of the neuron activation, $max(0, \mathbf{w} \cdot \mathbf{x} + b)$. The use of the $max(0, z)$ function call on scalar z just says to treat all negative z values as 0. The derivative of the max function is a piecewise function. When $z \leq 0$, the derivative is 0 because z is a constant. When $z > 0$, the derivative of the max function is just the derivative of z, which is 1:

$$\frac{\partial}{\partial z} max(0, z) = \begin{cases} 0 & z \leq 0 \\ \frac{dz}{dz} = 1 & z > 0 \end{cases}$$

An aside on broadcasting functions across scalars. When one or both of the *max* arguments are vectors, such as $max(0, \mathbf{x})$, we broadcast the single-variable function *max* across the elements. This is an example of an element-wise unary operator. Just to be clear:

$$max(0, \mathbf{x}) = \begin{bmatrix} max(0, x_1) \\ max(0, x_2) \\ \dots \\ max(0, x_n) \end{bmatrix}$$

For the derivative of the broadcast version then, we get a vector of zeros and ones where:

$$\frac{\partial}{\partial x_i} max(0, x_i) = \begin{cases} 0 & x_i \leq 0 \\ \frac{dx_i}{dx_i} = 1 & x_i > 0 \end{cases}$$

$$\frac{\partial}{\partial \mathbf{x}} max(0, \mathbf{x}) = \begin{bmatrix} \frac{\partial}{\partial x_1} max(0, x_1) \\ \frac{\partial}{\partial x_2} max(0, x_2) \\ \dots \\ \frac{\partial}{\partial x_n} max(0, x_n) \end{bmatrix}$$

To get the derivative of the $activation(\mathbf{x})$ function, we need the chain rule because of the nested subexpression, $\mathbf{w} \cdot \mathbf{x} + b$. Following our process, let's introduce intermediate scalar variable z to represent the affine function giving:

$$z(\mathbf{w}, b, \mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$$

$$activation(z) = max(0, z)$$

The vector chain rule tells us:

$$\frac{\partial activation}{\partial \mathbf{w}} = \frac{\partial activation}{\partial z} \frac{\partial z}{\partial \mathbf{w}}$$

which we can rewrite as follows:

$$\frac{\partial activation}{\partial \mathbf{w}} = \begin{cases} 0 \frac{\partial z}{\partial \mathbf{w}} = \vec{0}^T & z \leq 0 \\ 1 \frac{\partial z}{\partial \mathbf{w}} = \frac{\partial z}{\partial \mathbf{w}} = \mathbf{x}^T & z > 0 \quad \text{(we computed } \frac{\partial z}{\partial \mathbf{w}} = \mathbf{x}^T \text{ previously)} \end{cases}$$

and then substitute $z = \mathbf{w} \cdot \mathbf{x} + b$ back in:

$$\frac{\partial activation}{\partial \mathbf{w}} = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ \mathbf{x}^T & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

That equation matches our intuition. When the activation function clips affine function output z to 0, the derivative is zero with respect to any weight w_i. When $z > 0$, it's as if the *max* function disappears and we get just the derivative of z with respect to the weights.

Turning now to the derivative of the neuron activation with respect to b, we get:

$$\frac{\partial activation}{\partial b} = \begin{cases} 0 \frac{\partial z}{\partial b} = 0 & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ 1 \frac{\partial z}{\partial b} = 1 & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

Let's use these partial derivatives now to handle the entire loss function.

Training a neuron requires that we take the derivative of our loss or "cost" function with respect to the parameters of our model, $\mathbf{w}$ and b. Because we train with multiple vector inputs (e.g., multiple images) and scalar targets (e.g., one classification per image), we need some more notation. Let

$$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T$$

where $N = |X|$, and then let

$$\mathbf{y} = [target(\mathbf{x}_1), target(\mathbf{x}_2), \ldots, target(\mathbf{x}_N)]^T$$

where y_i is a scalar. Then the cost equation becomes:

$$C(\mathbf{w}, b, X, \mathbf{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - activation(\mathbf{x}_i))^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - max(0, \mathbf{w} \cdot \mathbf{x}_i + b))^2$$

Following our chain rule process introduces these intermediate variables:

$$\begin{aligned} u(\mathbf{w}, b, \mathbf{x}) &= max(0, \mathbf{w} \cdot \mathbf{x} + b) \\ v(y, u) &= y - u \\ C(v) &= \frac{1}{N} \sum_{i=1}^{N} v^2 \end{aligned}$$

Let's compute the gradient with respect to **w** first.

The gradient with respect to the weights

From before, we know:

$$\frac{\partial}{\partial \mathbf{w}} u(\mathbf{w}, b, \mathbf{x}) = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ \mathbf{x}^T & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

and

$$\frac{\partial v(y, u)}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}(y - u) = \vec{0}^T - \frac{\partial u}{\partial \mathbf{w}} = -\frac{\partial u}{\partial \mathbf{w}} = \begin{cases} \vec{0}^T & \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ -\mathbf{x}^T & \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$